APPROXIMATION AND INTERPOLATION EMPLOYING DIVERGENCE–FREE RADIAL BASIS FUNCTIONS WITH APPLICATIONS

A Dissertation

by

SVENJA LOWITZSCH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2002

Major Subject: Mathematics

APPROXIMATION AND INTERPOLATION EMPLOYING

DIVERGENCE–FREE RADIAL BASIS FUNCTIONS WITH APPLICATIONS

A Dissertation

by

SVENJA LOWITZSCH

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Francis J. Narcowich (Co-Chair of Committee)	Joseph D. Ward (Co-Chair of Committee)
Joseph E. Pasciak (Member)	Natarajan Sivakumar (Member)
Randall L. Eubank (Member)	William Rundell (Head of Department)

May 2002

Major Subject: Mathematics

ABSTRACT

Approximation and Interpolation Employing Divergence–free Radial Basis Functions with Applications. (May 2002)

Svenja Lowitzsch, Dipl., Georg-August University, Göttingen

Co–Chairs of Advisory Committee: Dr. Francis J. Narcowich
Dr. Joseph D. Ward

Approximation and interpolation employing radial basis functions has found important applications since the early 1980's in areas such as signal processing, medical imaging, as well as neural networks. Several applications demand that certain physical properties be fulfilled, such as a function being divergence free. No such class of radial basis functions that reflects these physical properties was known until 1994, when Narcowich and Ward introduced a family of matrix-valued radial basis functions that are divergence free. They also obtained error bounds and stability estimates for interpolation by means of these functions. These divergence-free functions are very smooth, and have unbounded support. In this thesis we introduce a new class of matrix-valued radial basis functions that are divergence free as well as compactly supported. This leads to the possibility of applying fast solvers for inverting interpolation matrices, as these matrices are not only symmetric and positive definite, but also sparse because of this compact support. We develop error bounds and stability estimates which hold for a broad class of functions. We conclude with applications to the numerical solution of the Navier-Stokes equation for certain incompressible fluid flows.

To my parents

ACKNOWLEDGMENTS

I would like to thank my parents, Uschi and Jürgen, for their endless love and support during all these years of studies. I would not have been able to go through all this without them.

A special thanks also to my advisors, Dr. Narcowich and Dr. Ward, who gave me this interesting and challenging problem and who helped me through my struggles with it.

I also would like to thank all my dear friends, and especially Antonio, Mikey, and Tatev for their open arms to give me a hug whenever I needed one and for their patience to listen to my problems.

And last, but not least, thanks to Monique, because she smoothed many hurdles for me.

LIST OF SYMBOLS

TABLE OF CONTENTS

LIST OF TABLES

TABLE Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Methods of approximation and interpolation employing radial basis functions (RBFs) go back to the 1970's. In 1971, Hardy developed so-called *multiquadrics* that have been applied intensively in surface fitting problems, e.g. in geophysics [10, 11]. In the late 1970's, Duchon derived a first variational formulation employing so-called *thin-plate splines* [4, 6, 5]. In 1982, Franke [8] posed the question of invertibility of the interpolation matrix resulting from the Hardy multiquadrics. In the mid 1980's, Madych and Nelson [17, 18] and Micchelli [19] obtained results answering the invertibility question. Further work concerning the variational formulation similar to the developments of Duchon was done by Madych and Nelson in [17, 18]. A good overview was done by Buhmann [2], Dyn [7], and Powell [24]. Interpolation and approximation via RBFs have been of interest because of their applications in signal processing, computer aided design, and computed tomography, as well as in other areas. Radial basis functions also have been used to obtain numerical solutions of partial differential equations (PDEs) resulting from physical problems. In 1990, Kansa introduced collocation methods for numerically solving PDEs [15, 16].

In the particular situation of interpolating data that stems from an incompressible fluid, i.e. a fluid having neither sources nor sinks, we say that the fluid has a velocity field $\mathbf{v}(x)$ that is *divergence-free* if

$$\nabla \cdot \mathbf{v} \equiv 0. \tag{1.1}$$

So far, interpolation of data originating from a fluid has been done via classical,

---

The journal model is Mathematics of Computation.

scalar-valued RBFs, e.g. the *Gaussians*, $\Phi(x) = e^{-t\|x\|^2}$, $t > 0$, or Hardy multiquadrics, see [12]. In many situations it is important for the interpolant to reflect the physical behavior of the data, and this means being divergence-free for incompressible fluids. But interpolants resulting from classical RBFs do not have this property. In 1994, Narcowich and Ward [22] constructed a matrix-valued RBF whose columns are divergence-free vector fields, which then gives rise to divergence-free interpolants. These functions are generated by smooth RBFs of unbounded support. This results in rather expensive calculations.

In this thesis, we present a new class of divergence-free RBFs, which are based on compactly-supported scalar-valued RBFs, and hence become much more efficient for computational purposes, since fast solvers can be applied. Other desirable physical properties can also be fulfilled, for example a function being curl free. However, the theory established in [22], as well as in this thesis is not restricted to functions that are divergence free. Therefore, customized basis functions can be constructed. Up to this thesis, only one example of stability bounds [22], no error estimates, and no applications had been investigated for this new class of functions.

Given an s-dimensional divergence-free vector field on $\mathbb{R}^s$, the general form of an $s \times s$ matrix-valued RBF can be written as

$$\Phi(x) = \{-\Delta I + \nabla\nabla^T\}\psi(x), \tag{1.2}$$

where ψ is a scalar-valued, compactly-supported RBF, Δ is the Laplacian operator, ∇ is the gradient, and I is the s-dimensional identity matrix. The differential operator $\{-\Delta I + \nabla\nabla^T\}$ is the same as introduced in [22], but we expand the approach to the new class of functions whose properties are very different from those in [22]. As compactly-supported RBFs we use the recently discovered *Wendland functions*, introduced in [27], which can be constructed to be C^{2k}-functions for any desired level

k of smoothness.

Let Ω be a compact subset of $\mathbb{R}^s$. In the case where the interpolating data appears in the form $\{x_j, \mathbf{d}_j\}_{i=1}^N$, where $\mathbf{d} = \{\mathbf{d}_j\}_{j=1}^N \in \mathbb{R}^s$ are data sites stemming from a vector-valued function $\mathbf{f}$ at given points $X = \{x_j\}_{j=1}^N \in \Omega$, the divergence-free interpolant becomes

$$\mathbf{s}_{\mathbf{f},X}(x) = \sum_{j=1}^{N} \Phi(x - x_j)\mathbf{c}_j. \tag{1.3}$$

The interpolation problem is to find $\{c_j\}_{j=1}^N$ such that

$$\mathbf{s}_{\mathbf{f},X}(x_k) = \mathbf{d}_\mathbf{k} \text{ for all } 1 \leq k \leq N. \tag{1.4}$$

This problem can be restated as a system of linear equations, with the resulting interpolation matrix being positive definite, symmetric, and sparse. Hence, fast solvers can be applied to obtain a unique solution. This motivates further research of the new class of functions.

In Chapter II we state the necessary definitions and introduce notation. We define admissible spaces and introduce $\mathcal{S}$-conditionally positive matrix-valued radial basis functions. Since Fourier transforms and convolutions play an important role in this thesis, we give their definitions. We also define the *native space* of a radial basis function. We then introduce the generalized Hermite interpolation problem, and state some important results connected with it. We conclude the chapter by discussing Sobolev spaces and related aspects.

Chapter III introduces the new class of divergence-free matrix-valued functions of compact support. We first give a background for the compactly-supported Wendland functions. We then define the new class of functions, and show that they have the following properties: being divergence-free, having compact support, as well as being strictly positive definite. We terminate this chapter by proving directly that the smooth matrix-valued RBFs introduced in [22] are divergence-free and strictly positive definite.

In Chapter IV we derive a density theorem that guarantees the existence of a divergence-free interpolant of type (1.3) that approximates any given divergence-free function arbitrarily well. This result can be viewed as a kind of Weierstrass Approximation Theorem involving matrix-valued divergence-free RBFs of compact support. Instead of an approximation of a continuous function by polynomials, we here approximate a divergence-free continuous function by a linear combination of divergence-free compactly-supported functions. In the case of scalar-valued RBFs, a density result was derived, the so-called Brown's Theorem [24]. Since our functions have the additional property of being divergence-free we use a different technique to obtain a density result. We start with some preliminaries, state and prove the density theorem, and end the chapter with several remarks.

Chapter V presents rates of approximation concerning the matrix-valued RBF interpolation and approximation problem. Error estimates measure the worst deviation of the interpolant from the function generating the data. The error estimates obtained here are comparable to the scalar-valued results stated in [23]. They are of the form

$$\sup_{x \in W(\omega,\delta) \subset \Omega} \|D^\alpha (f - s_f)(x)\|_\infty \le |f|_\Phi \, C \, h_{X,\Omega}^\beta \tag{1.5}$$

for some $\beta = \beta(\alpha, \Phi)$, where $h_{X,\Omega} := \sup_{y \in \Omega} \min_{x_j \in X} \|x_j - y\|_2$ is the so-called *mesh norm* and C is a constant independent of f and N. The upper bound depends on f and the mesh norm as desired. We start this chapter with some preliminary error estimates, investigate the so-called *power function* arising in conjunction with them. (The power function is the main term of the error estimates that needs to be bounded). We then discuss *norming sets* [13] which are used to obtain bounds for the power function. Combining all these results, we obtain error estimates for the interpolation problem of the form (1.5). We end the chapter by deriving error estimates for specific functions.

In Chapter VI we investigate the stability of the interpolation matrix based on the matrix-valued RBF, and this is done via a study of condition numbers. Stability determines how much the interpolant changes in case of (small) perturbations of the data. Since real data usually has some error arising from the method of its measurement, stability estimates are of profound importance. The results reflect the expected behavior obtained for the special case of the Gaussian function in [22]. We investigate the norm of the inverse of the interpolation matrix A. The result is of the form

$$\|A^{-1}\| \leq \theta^{-1}, \tag{1.6}$$

where $\theta = \theta(q, s, p)$ depends only on the space dimension s, the *separation distance*, q, which is defined by the equation $2q := \min_{1\leq j\neq k\leq N} \|x_j - x_k\|_2$, as well as on a certain integer p. Since the parameter p can always be determined from s, the upper bound actually only depends on the separation distance and the space dimension, as desired. In this chapter we first introduce the multivariate Hermitian interpolation problem, where the data comes from point evaluations, as well as from derivative information. We then obtain upper bounds for the quadratic form arising from the Hermitian problem. In this analysis, we bound the matrix-valued RBF by a product of characteristic functions χ whose support is the cube with edges $[-\pi/2, \pi/2]$ in each direction. We then obtain some upper bounds on the derivatives of χ. This leads to the aforesaid stability estimates, which are first stated and proved, and subsequently applied to several special functions.

In Chapter VII we apply the methods derived in this thesis to problems based on fluids that are described by the incompressible Navier-Stokes equations. In these applications we use MATLAB as a visualization tool. We study the *driven cavity problem*, where a rectangular cavity is given with a horizontal air flow at the top. The objective is to reproduce the air flow inside the cavity when the flow reaches its steady state. We first give a description of the general driven cavity problem. We then investigate time-independent, constant-pressure problems based on cavities of different shapes which result in non-linear PDEs with given boundary values. We use an iterative algorithm to solve these problems. We describe the setup of the problems, i.e. we determine the variables and the scalar-valued RBF. We then give error results, orders of approximation, and figures to visualize the results obtained. These results are physically reasonable. We conclude the chapter with an investigation of a time-independent but general-pressure problem. We first describe the PDE and the boundary values. We then derive an algorithm that implements a simultaneous interpolation of the fluid flow and the pressure. This algorithm has not yet been implemented, but is of interest for future research.

The thesis comes to a close with Chapter VIII, where we summarize our studies and state directions for future research.

CHAPTER II

NOTATION AND DEFINITIONS

In this chapter we introduce notation and definitions necessary for interpolation with matrix-valued radial basis functions (RBFs).

A. General Notation

Let $\mathbb{P}_{\mathrm{m}}^{\mathrm{d}}$ be the space of $\mathbb{C}$-valued polynomials of order not exceeding m in $\mathbb{R}^{\mathrm{d}}$. Further, let $\mathbb{P}_{\mathrm{m}}^{\mathrm{d}\mapsto\mathrm{s}}$ denote the space of $\mathbb{C}^s$-valued polynomials of order not exceeding m defined on $\mathbb{R}^{\mathrm{d}}$. If $s = 1$, then $\mathbb{P}_{\mathrm{m}}^{\mathrm{d}\mapsto 1} = \mathbb{P}_{\mathrm{m}}^{\mathrm{d}}$.

Define the multi-index notation as usual: Let $\alpha = (\alpha_1, \ldots, \alpha_d)$ be a d-tuple of nonnegative integers, define $|\alpha| = \sum_j \alpha_j$, $\alpha! := \alpha_1! \cdots \alpha_d!$, $t^\alpha := t(1)^{\alpha_1} \cdots t(d)^{\alpha_d}$, and so on. We use $t(j)$ to denote the j^{th} component of $t \in \mathbb{R}^{\mathrm{d}}$.

We define the *floor function* to be the function $\lfloor x \rfloor$ which returns the integer k such that $k - 1 \leq x < k$ and the *ceiling function* to be $\lceil x \rceil = \lfloor x \rfloor + 1$, for $x \in \mathbb{R}_{\geq 0}$.

Let $C_\nu^k(\mathbb{R}^{\mathrm{d}})$ be the space of functions in $C^k(\mathbb{R}^{\mathrm{d}})$ that have all kth order derivatives Hölder continuous at the origin, with Hölder exponent $0 < \nu \leq 1$. We denote its norm by $\|\cdot\|_{C_\nu^k}$.

B. Admissible Spaces

Define $\mathcal{E}_s$ to be the set of all C^∞ vector-valued functions $f : \mathbb{R}^{\mathrm{d}} \to \mathbb{C}^{\mathrm{s}}$, and denote $\mathcal{E}_s'$ to be the set of all s-valued distributions of compact support in $\mathbb{R}^{\mathrm{d}}$. If $\mu \in \mathcal{E}_s'$, then let $\mu^* := \bar{\mu}^T$ be the conjugate transpose of μ. The *linear functional* corresponding to the distribution μ acts on $f \in \mathcal{E}_s$ via

$$(\mu^*, f) = \int_{\mathbb{R}^d} \mu(x)^* f(x)\, dx = \sum_{j=1}^{s} \int_{\mathbb{R}^d} \bar{\mu}_j(x) f_j(x)\, dx,$$

where the μ_j's and the f_j's are the components of μ and f, respectively.

Fourier transforms and convolutions play an important role in interpolation employing RBFs. We define the *Fourier transform* of a function or distribution to be

$$\hat{f}(\omega) := \int\limits_{\mathbb{R}^d} e^{-ix\cdot\omega} f(x)\, dx,$$

and let the *inverse Fourier transform* of a function or distribution be defined by

$$\check{g}(x) := (2\pi)^{-d} \int\limits_{\mathbb{R}^d} e^{ix\cdot\omega} g(\omega)\, d\omega.$$

Note that the Fourier transform of a compactly-supported distribution is an analytic function. For a scalar-valued distribution μ and function $f \in \mathcal{E}_s$, we define the *scalar-valued convolution* $f * \mu$ component-wise, with the j^{th} component being given by the expression $[f * \mu(x)]_j = \int_{\mathbb{R}^d} f_j(x-y)\mu(y)\, dy$. For a vector-valued distribution μ and an $s \times s$ matrix H with each component in C^∞, we define the *vector-valued convolution* $H * \mu$ component-wise as well, with the j^{th} component being given by the expression $[H * \mu(x)]_j = \sum_{k=1}^{s} \int_{\mathbb{R}^d} H_{j,k}(x-y)\mu_k(y)\, dy$, where the $H_{j,k}$'s and the μ_k's are the components of H and μ, respectively.

In order to incorporate side conditions on functions in $\mathcal{E}_s$ we work with certain subspaces of $\mathcal{E}_s$ which we now define.

Definition 2.1. Let $\mathcal{S}$ be a subspace of $\mathcal{E}_s$ such that if g is in $\mathcal{S}$ and μ is an arbitrary scalar-valued distribution in $\mathcal{E}_1'$, then $g*\mu$ is in $\mathcal{S}$. We shall call such a space *admissible.*

For much of the thesis, we consider admissible spaces arising as kernels of differential operators of constant coefficients.

Definition 2.2. For an integer $\nu \geq 1$, let $B_1(x), \ldots, B_\nu(x)$ be $\mathbb{C}^s$-valued polynomials defined on $\mathbb{R}^d$ having degree n component-wise. Define the space

$$\mathcal{S}_B := \{f \in \mathcal{E}_s : B_j(\nabla)^* f \equiv 0,\ 1 \leq j \leq \nu\}, \quad \text{where } \nabla := (\partial_{x(1)}, \ldots, \partial_{x(d)})^T.$$

These spaces are admissible because $B_j(\nabla)^*(f * \mu) = (B_j(\nabla)^* f) * \mu = 0$. See also [26]. Note that when $\nu = 1$, $s = 3$, and $B_1(x) = (x_1, x_2, x_3)^T$, then $\mathcal{S}_B = \mathcal{S}_{\text{div}}$, the admissible space of divergence-free vector-valued functions.

C. $\mathcal{S}$-linearly Independent Distributions

We assume that the data is generated by applying a finite number of compactly-supported distributions to a function $f \in \mathcal{E}_s$. More precisely, let $\Lambda := \{\lambda_j\}_{j=1}^N$ be a linearly independent subset of $\mathcal{E}_s'$, and assume that we are given data in the form

$$(\lambda_j^*, f) = d_j \text{ for } 1 \leq j \leq N, \tag{2.1}$$

where the d_j's are scalars for $1 \leq j \leq N$. To avoid redundant data, we require that the distributions generating the data be linearly independent when restricted to the space $\mathcal{S}$. We denote the set $\Lambda = \{\lambda_j\}_{j=1}^N$ to be $\mathcal{S}$-*linearly independent* if the set $\Lambda|_{\mathcal{S}} = \{\lambda_j|_{\mathcal{S}}\}_{j=1}^N$ is linearly independent.

In order to be able to deal with interpolation problems requiring polynomial reproduction, we introduce the following class of subspaces of $\mathcal{E}_s'$:

$$\mathcal{E}_{s,m}'(\mathcal{S}) := \{\mu \in \mathcal{E}_s' : (\mu^*, p) = 0 \text{ for all } p \in \Pi_m^{d \mapsto s} \cap \mathcal{S}\}, \tag{2.2}$$

if $m \geq 1$. In case $m = 0$, we set $\mathcal{E}_{s,0}' := \mathcal{E}_s'$. When $\mathcal{S} = \mathcal{E}_s$, we write $\mathcal{E}_{s,m}'$ for $\mathcal{E}_{s,m}'(\mathcal{S})$. From (2.2), note that $\mathcal{E}_{s,m}' \subseteq \mathcal{E}_{s,m}'(\mathcal{S})$.

D. Conditionally Positive Definite Functions

We now define the class of functions that are used to generate the interpolants. Given a vector-valued distribution $\mu \in \mathcal{E}'_s$ and an $s \times s$ matrix Φ with each entry being in $C^\infty(\mathbb{R}^d)$, we define

$$(\mu^* \otimes \mu, \Phi) := \int_{\mathbb{R}^d} \mu(x)^* (\Phi * \mu)(x)\, dx. \tag{2.3}$$

Definition 2.3. Let Φ be an $s \times s$ matrix whose components are in $C^\infty(\mathbb{R}^s)$ and whose columns are in an admissible space $\mathcal{S}$. Require also that $\Phi(x)^* = \Phi(-x)$ for all $x \in \mathbb{R}^d$. We say that Φ is an *(order-m) $\mathcal{S}$-conditionally positive definite* ($\mathcal{S}$-CPD) *$s \times s$ matrix-valued function* if

$$(\mu^* \otimes \mu, \Phi) \geq 0 \text{ for all } \mu \in \mathcal{E}'_{s,m}(\mathcal{S}). \tag{2.4}$$

If equality in (2.4) implies that $(\mu^*, g) = 0$ for all $g \in \mathcal{S}$, we say that Φ is *strictly* $\mathcal{S}$-CPD.

When $m = 0$, the distributions used in (2.4) are independent of $\mathcal{S}$. In this case we say that Φ is *strictly positive definite* (SPD). If Φ is a SPD matrix, the sesquilinear form

$$\langle \eta, \lambda \rangle_\Phi := (\lambda^* \otimes \eta, \Phi) \tag{2.5}$$

defines an inner product for the space $\mathcal{E}'_s$.

In case of no side conditions, i.e. $\mathcal{S} = \mathcal{E}_s$, we say that the distributions are *conditionally positive definite* (CPD) rather than $\mathcal{S}$-CPD. Note that $\mathcal{E}'_{s,m} \subseteq \mathcal{E}'_{s,m}(\mathcal{S})$ implies that every order-m $\mathcal{S}$-CPD distribution is also order-m CPD. Given a strictly

$\mathcal{S}$-CPD matrix-valued function Φ, we can define an inner product on $\mathcal{E}'_{s,m}(\mathcal{S})$:

$$(2.6) \qquad \langle \mu, \lambda \rangle_\Phi := (\lambda^* \otimes \mu, \Phi), \quad \mu, \lambda \in \mathcal{E}'_{s,m}(\mathcal{S}).$$

If $m = 0$, the expression (2.6) agrees with (2.5). The norm for this inner product is denoted by $\|\cdot\|_\Phi$.

Note that the inner product spaces associated with Φ are defined for distributions, rather than functions, and are usually not complete. If we look at the functions themselves, we can define an associated norm for functions. Let $f = \Phi * \nu + p$, with $\nu \in \mathcal{E}'_{s,m}(\mathcal{S})$ and $p \in \mathbb{P}_{\mathrm{m}}^{\mathrm{s}\mapsto\mathrm{s}} \cap \mathcal{S}$. Obviously, if $\mu \in \mathcal{E}'_{s,m}(\mathcal{S})$, then $(\mu^*, f) = \langle \nu, \mu \rangle_\Phi$. We define the associated norm for f to be

$$(2.7) \qquad |f|_\Phi^2 := \sup_{\|\mu\|_\Phi = 1} |(\mu^*, f)|.$$

Clearly, if f has the decomposition $f = \Phi * \nu + p$, then we have $|f|_\Phi = \|\nu\|_\Phi$. Equation (2.7) also defines the norm of f in the completion of the inner product space. It leads to a definition of the *native space* of Φ:

$$(2.8) \qquad \mathcal{N}_\Phi := \left\{ f \in C^\infty(\mathbb{R}^{\mathrm{s}}; \mathbb{C}^{\mathrm{s}}) : \sup_{\|\mu\|_\Phi = 1} |(\mu^*, f)| < \infty \right\},$$

where $f \in C^\infty(\mathbb{R}^{\mathrm{s}}; \mathbb{C}^{\mathrm{s}})$ means that each component of the s-variant function f is in $C^\infty(\mathbb{R}^{\mathrm{s}})$. The norm of the native space is comparable to the norm of a a reproducing kernel Hilbert space. The native space guarantees that the application of linear functionals to a function f is well-defined.

E. Generalized Hermite Interpolation Problem

Hardy introduced the Hermite interpolation problem in 1990 [11]. One can use a strictly $\mathcal{S}$-CPD matrix-valued function Φ to solve the following generalized Hermite interpolation problem.

Problem 2.4. Generalized Hermite Interpolation Problem. Assume that Φ is a strictly order-m $\mathcal{S}$-CPD, $s \times s$ matrix-valued function. Let $\Lambda = \{\lambda_1, \dots \lambda_N\}$ be a $\mathcal{S}$-linearly independent set of distributions on $\mathcal{E}'_s$, and let f be a function in $\mathcal{E}_s$. Given the data $d_j = (\lambda_j^*, f)$, for $j = 1, \dots, N$, we seek to find $\lambda \in \mathrm{span}\{\Lambda\} \cap \mathcal{E}'_{s,m}(\mathcal{S})$ and $p \in \mathrm{I\!P}_{\mathrm{m}}^{\mathrm{s} \mapsto \mathrm{s}} \cap \mathcal{S}$ such that $\Phi * \lambda \in \mathcal{S}$ and $s_f = \Phi * \lambda + p$ satisfies

$$(\lambda_j^*, s_f) = d_j, \text{ for } j = 1, \dots, N, \tag{2.9}$$

and, if f is in $\mathrm{I\!P}_{\mathrm{m}}^{\mathrm{s} \mapsto \mathrm{s}} \cap \mathcal{S}$, we require that $s_f = p = f$.

Narcowich and Ward proved in [22] the following result which provides the framework for the requirement that the interpolation problem 2.4 reproduces polynomials.

Theorem 2.5. *Let* $\mathcal{U} = \mathrm{span}\{\Lambda\}$ *and* $\mathcal{W} = \mathcal{U} \cap \mathcal{E}'_{s,m}(\mathcal{S})$. *If the dimension of* $\mathcal{U} \setminus \mathcal{W}$ *is equal to the dimension* m' *of* $\mathrm{I\!P}_{\mathrm{m}}^{\mathrm{d} \mapsto \mathrm{s}} \cap \mathcal{S}$, *then Problem 2.4 is well-poised.*

As a byproduct of the proof of Theorem 2.5, the following result is obtained in [22].

Proposition 2.6. *Let* $\Lambda = \{\lambda_1, \dots \lambda_N\}$ *be a* $\mathcal{S}$*-linearly independent set of the space* $\mathcal{E}'_{s,m}(\mathcal{S})$. *If* Φ *is an* $s \times s$ *matrix-valued, order-*m $\mathcal{S}$*-CPD function, then the interpolation matrix* A *with entries*

$$A_{j,k} = (\lambda_k^* \otimes \lambda_j, \Phi), \text{ for } 1 \le j, k \le N, \tag{2.10}$$

is selfadjoint and nonnegative. If Φ *is strictly* $\mathcal{S}$*-CPD, then the matrix* A *is positive definite.*

F. Sobolev Spaces

In this thesis we need various vector-valued functions $f = (f_1, \dots, f_s)^T$ to be in certain Sobolev spaces. Let Ω be an open subset of $\mathbb{R}^{\mathrm{d}}$. In particular, Ω can be the whole space $\mathbb{R}^{\mathrm{d}}$. We define

$$(2.11)\quad H^k(\Omega;\, \mathbb{C}^{\mathrm{s}}) := clos\Big\{ f : \Omega \to \mathbb{C}^{\mathrm{s}} \text{ such that } \sum_{|\alpha|\le k} \sum_{i=1}^{s} \int_{\Omega} |D^\alpha f_i(x)|^2 \, dx < \infty \Big\}$$

with inner product

$$\langle f, g\rangle_{H^k(\Omega;\, \mathbb{C}^{\mathrm{s}})} = \sum_{|\alpha|\le k} \sum_{i=1}^{s} \int_{\Omega} \overline{D^\alpha f_i(x)} D^\alpha g_i(x) \, dx$$

for all $f = (f_1, \dots, f_s)^T$ and $g = (g_1, \dots, g_s)^T$ in $H^k(\Omega;\, \mathbb{C}^{\mathrm{s}})$. In case Ω is the whole $\mathbb{R}^{\mathrm{d}}$, we use Fourier transform arguments to derive an equivalent definition of $H^k(\mathbb{R}^{\mathrm{d}};\, \mathbb{C}^{\mathrm{s}})$ given by

$$H^k(\mathbb{R}^{\mathrm{d}};\, \mathbb{C}^{\mathrm{s}}) = clos\Big\{ f : \mathbb{R}^{\mathrm{d}} \to \mathbb{C}^{\mathrm{s}} \text{ such that } \int_{\mathbb{R}^{\mathrm{d}}} (1 + \|\xi\|^2)^k \|\hat{f}(\xi)\|^2 \, d\xi < \infty \Big\}.$$

We work with both definitions. We denote $H^k := H^k(\mathbb{R}^{\mathrm{d}};\, \mathbb{C}^{\mathrm{s}})$. The inner product of $H^k(\mathbb{R}^{\mathrm{d}};\, \mathbb{C}^{\mathrm{s}})$ is given by

$$\langle f, g\rangle_{H^k} = \int_{\mathbb{R}^{\mathrm{d}}} (1 + \|\xi\|^2)^k \hat{f}(\xi)^* \hat{g}(\xi) \, d\xi$$

for all f and g in $H^k(\mathbb{R}^{\mathrm{d}};\, \mathbb{C}^{\mathrm{s}})$.

The *dual* of H^k is given by H^{-k}. We define the Sobolev space of s-variate linear functionals which fulfill the side condition $\mathcal{S}$ to be

$$H^{-k}_m(\mathcal{S}) := \Big\{ \lambda \in H^{-k} \text{ such that } \int_{\mathbb{R}^{\mathrm{d}}} \lambda(x)^* p(x) \, dx = 0 \text{ for all } p \in \mathbb{P}^{\mathrm{d}\mapsto\mathrm{s}}_{\mathrm{m}} \cap \mathcal{S} \Big\}.$$

If $m = 0$, we set $H^{-k} := H^{-k}_0(\mathcal{S})$, since for $m = 0$, the set of polynomials $\mathbb{P}^{\mathrm{d}\mapsto\mathrm{s}}_{\mathrm{m}}$ is empty.

In the case of scalar-valued functions $f : \Omega \to \mathbb{C}$, we denote $H^k(\Omega) := H^k(\Omega; \mathbb{C})$. If Ω is the whole $\mathbb{R}^{\mathrm{d}}$, we again denote $H^k := H^k(\mathbb{R}^{\mathrm{d}})$. Which space we refer to is determined by the context.

The space of square-integrable vector-valued functions is defined as follows. Let Ω be an open subset of $\mathbb{R}^{\mathrm{d}}$. Then

$$L_2(\Omega; \mathbb{C}^{\mathrm{s}}) := \Big\{ f : \Omega \to \mathbb{C}^{\mathrm{s}} \text{ such that } \int_{\Omega} \|f(x)\|_2^2 \, dx < \infty \Big\}. \tag{2.12}$$

In the case that Ω is the whole $\mathbb{R}^{\mathrm{d}}$, we denote $L_2 := L_2(\mathbb{R}^{\mathrm{d}}; \mathbb{C}^{\mathrm{s}})$.

CHAPTER III

DIVERGENCE-FREE MATRIX-VALUED FUNCTIONS

In this chapter we will investigate classes of divergence-free matrix-valued functions which are strictly positive definite. We derive a new class of divergence-free C^{2k} matrix-valued functions that are compactly supported. We conclude the chapter with the study of a divergence-free matrix-valued C^∞ function, introduced in [22], which is based on the Gaussian function.

A. Divergence-free Compactly-supported Functions

We introduce a new class of divergence-free compactly-supported functions. These functions are based on the *Wendland functions* $\psi_{l,k}$ introduced in [27]. We begin by stating important characteristics of the Wendland functions and then introduce the new class of functions.

1. Wendland Functions

We follow [27] in defining the Wendland functions to be

$$\psi_{l,k}(r) := \mathcal{I}^k \psi_l(r) = \mathcal{I}^k (1-\cdot)_+^l(r),$$

where

$$\mathcal{I}\ \psi(r) := \begin{cases} \int_r^\infty s\psi(s)\ ds & \text{for } r > 0, \\ \mathcal{I}\ \psi(-r) & \text{for } r < 0, \end{cases}$$

with $k \in \mathbb{N}_0$, $l \in \mathbb{R}_{>0}$, and the cut-off function $(1-r)_+^l$ is defined as

$$(1-r)_+^l := \begin{cases} (1-r)^l & \text{for } 0 \le r \le 1, \\ 0 & \text{for } r \ge 1. \end{cases}$$

Note that $r = \|x\|$ for all the Wendland functions. Here, k is related to the smoothness and l to the degree of the Wendland function. If $k \in \mathbb{N}$ and $l = \lfloor d/2 \rfloor + k + 1$, where d is the space dimension, then $\psi_{l,k}$ is in $\mathrm{SPD} \cap C^{2k}$, and $\psi_{l,k}$ is of degree $\lfloor d/2 \rfloor + 3k + 1$, see [27, Theorem 3.1]. As an example, Figure 1 shows the function $\psi_{4,2}(r) \doteq (1-r)_+^6(35r^2 + 18r + 3) \in C^4$, where $\doteq$ means equality up to a constant. Table 1 contains a list of several Wendland functions. It is taken from [27].

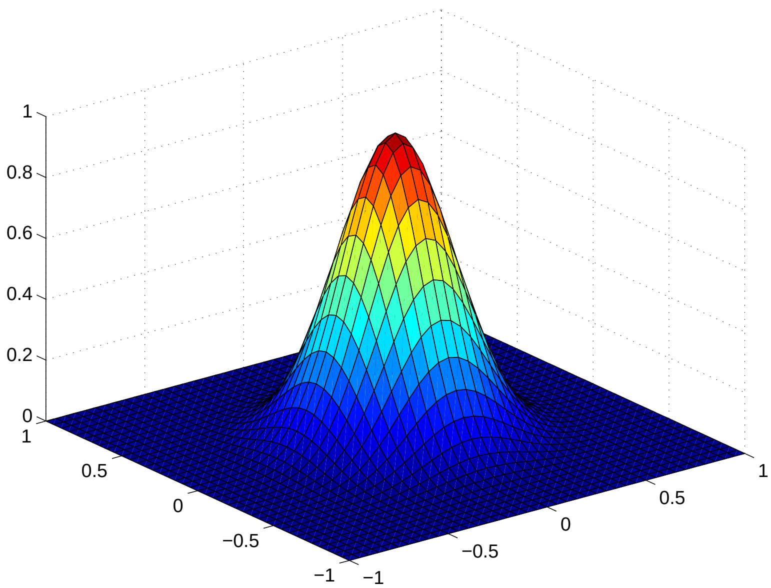

FIGURE 1. $\psi_{4,2}(r) \doteq (1-r)_+^6(35r^2 + 18r + 3)$

Lemma 3.1. *For every space of dimension $d \in \mathbb{N}$ and every $k \in \mathbb{N}$, set the integer $l = \lfloor d/2 \rfloor + k + 1$. Then the function $\psi_{l,k}$ is in H^{2k}.*

Proof. Firstly, observe that $l = \lfloor d/2 \rfloor + k + 1$ implies $\psi_{l,k} \in C^{2k}$, and hence $|D^\alpha \psi_{l,k}|^2$ is continuous for all $|\alpha| \leq 2k$. Secondly, since $\psi_{l,k}$ is compactly-supported, $|D^\alpha \psi_{l,k}|^2$ is also compactly-supported for all $|\alpha| \leq 2k$. Hence, we conclude that

$$\sum_{|\alpha| \leq 2k} \int_{\mathbb{R}^d} |D^\alpha \psi_{l,k}(x)|^2 \, dx < \infty,$$

Table 1. Several Wendland functions

d=1	$\psi_{1,0}(r) = (1-r)_+$	C^0
	$\psi_{2,1}(r) \doteq (1-r)_+^3(3r+1)$	C^2
	$\psi_{3,2}(r) \doteq (1-r)_+^5(8r^2+5r+1)$	C^4
d=3	$\psi_{2,0}(r) = (1-r)_+^2$	C^0
	$\psi_{3,1}(r) \doteq (1-r)_+^4(4r+1)$	C^2
	$\psi_{4,2}(r) \doteq (1-r)_+^6(35r^2+18r+3)$	C^4
	$\psi_{5,3}(r) \doteq (1-r)_+^8(32r^3+25r^2+8r+1)$	C^6
d=5	$\psi_{3,0}(r) = (1-r)_+^3$	C^0
	$\psi_{4,1}(r) \doteq (1-r)_+^5(5r+1)$	C^2
	$\psi_{5,2}(r) \doteq (1-r)_+^7(16r^2+7r+1)$	C^4

which yields that $\|\psi_{l,k}\|_{2k}$ is bounded, and therefore, $\psi_{l,k}$ is in H^{2k}. $\square$

2. Construction of New Functions

In this section we derive a new class of divergence-free matrix-valued radial basis functions of compact support which are based on the Wendland functions that live in certain Sobolev spaces $H^k(\mathbb{R}^d)$. In [22, Theorem 3.2] a class of divergence-free functions is introduced which are in C^∞. We now focus on compactly-supported functions, since they have the advantage of providing sparse interpolation matrices and hence result in much faster computations.

Consider the space of dimension $d = s$, since we have square matrix-valued radial basis functions. Let $\Pi(\xi) := I - \|\xi\|^{-2}\xi\xi^T$ for all $\xi \in \mathbb{R}^s$ such that $\xi \neq 0$. Choose the measure $d\mu_{l,k}$ to be

$$d\mu_{l,k}(\xi) := (2\pi)^{-s}\|\xi\|^2\Pi(\xi)\hat{\psi}_{l,k}(\xi)\, d\xi, \tag{3.1}$$

where $\psi_{l,k}$ is a scalar-valued Wendland function on $\mathbb{R}^s$. We now show that the function $\Phi_{l,k}(x) := \int\limits_{\mathbb{R}^d} e^{ix\cdot\xi}\, d\mu_{l,k}(\xi)$ is strictly positive definite.

Theorem 3.2. *Let* $d\mu_{l,k}(\xi) := (2\pi)^{-s}\|\xi\|^2\Pi(\xi)\hat{\psi}_{l,k}(\xi)\, d\xi$*, where* $\psi_{l,k}$ *is a scalar-valued Wendland function for* $l \geq 2$ *and* $k \geq 1$*. Then* $d\mu_{l,k}$ *is a selfadjoint and positive* $s \times s$ *matrix-valued Borel measure defined on* $\mathbb{R}^s$ *satisfying* $B(-i\xi)^* d\mu_{l,k}(\xi) \equiv 0$ *with* $B(x) := x$ *as introduced in Definition 2.2, and*

$$\Phi_{l,k}(x) := \int\limits_{\mathbb{R}^d} e^{ix\cdot\xi}\, d\mu_{l,k}(\xi) \tag{3.2}$$

is a strictly positive definite, compactly-supported $s \times s$ *matrix-valued function.*

Proof. Clearly, $d\mu_{l,k}$ is a selfadjoint, $s\times s$ matrix-valued Borel measure defined on $\mathbb{R}^s$ satisfying $B(-i\xi)^* d\mu_{l,k}(\xi) \equiv 0$ with $B(x) = x$. Since $\psi_{l,k}$ is of compact support, its Fourier transform $\hat{\psi}_{l,k}$ is analytic and decays exponentially. The positiveness of the measure follows from the fact that

$$\hat{\psi}_{l,k}(t) = t^{-s-2k-l}\int_0^t (t-r)^l r^{\frac{s}{2}+k} J_{\frac{s}{2}+k-1}(r)\, dr$$

is strictly positive, except for $\psi_{1,0}$, as was shown in [27]. We now show that $\Phi_{l,k}(x)$ is of compact support. Note that the inverse Fourier transform yields for $\xi(k)$, the k^{th}

coordinate of ξ,

$$\begin{aligned}
(i\xi(k)\hat{f}(\xi))^{\vee}(\omega) &= (2\pi)^{-s}\int\limits_{\mathbb{R}^s} e^{i\omega\cdot\xi}\, i\xi(k)\ \hat{f}(\xi)\ d\xi = (2\pi)^{-s}\int\limits_{\mathbb{R}^s} i\xi(k)\ e^{i\omega\cdot\xi}\hat{f}(\xi)\ d\xi \\
&= (2\pi)^{-s}\int\limits_{\mathbb{R}^s} \partial_{\omega(k)}\ e^{i\omega\cdot\xi}\hat{f}(\xi)\ d\xi = \partial_{\omega(k)}\ (2\pi)^{-s}\int\limits_{\mathbb{R}^s} e^{i\omega\cdot\xi}\hat{f}(\xi)\ d\xi \\
&= \partial_{\omega(k)} f(\omega)
\end{aligned}$$

for a scalar-valued function f, and hence, in general,

$$((i\xi)^{\alpha}\hat{f}(\xi))^{\vee}(\omega) = D^{\alpha} f(\omega) \text{ for all } |\alpha| \geq 0. \tag{3.3}$$

We use equation (3.3) to calculate $\Phi_{l,k}$ directly. We obtain:

$$\begin{aligned}
\Phi_{l,k}(x) &= (2\pi)^{-s}\int\limits_{\mathbb{R}^s} e^{ix\cdot\xi}\{\|\xi\|^2 I - \xi\xi^T\}\ \hat{\psi}_{l,k}(\xi)\ d\xi \\
&= [\{\|\xi\|^2 I - \xi\xi^T\}\ \hat{\psi}_{l,k}(\xi)]^{\vee}(x) \\
&= [\{-\|i\xi\|^2 I + (i\xi)(i\xi)^T\}\ \hat{\psi}_{l,k}(\xi)]^{\vee}(x) \\
&= \{-\Delta I + \nabla\nabla^T\}\ \psi_{l,k}(x),
\end{aligned}$$

where the *Laplacian operator* is defined to be $\Delta := \sum_{i=1}^{s} \partial^2_{x(i)}$ and the *gradient* is given as $\nabla := (\partial_{x(1)}, \ldots, \partial_{x(s)})^T$. Therefore, the components of $\Phi_{l,k}(x)$ are combinations of second order derivatives of $\psi_{l,k}(x)$, and hence every component of the matrix $\Phi_{l,k}(x)$ is in $C^{2k-2}(\mathbb{R}^s)$ and has compact support. Similar to Lemma 3.1 we obtain that every component of $\Phi_{l,k}$ is in H^{2k-2}.

We conclude the proof by showing that $\Phi_{l,k}$ is strictly positive definite. Firstly, it is clear that $\Phi_{l,k}(x)^* = \Phi_{l,k}(-x)$ for all $x \in \mathbb{R}^s$. Secondly, setting $m := k-1$ and defining $\Phi^{i,j}_{l,k}(x)$ to be the ij^{th} element of $\Phi_{l,k}(x)$, we get that $\Phi^{i,j}_{l,k}(x) \in H^{2m}$ for all $1 \leq i,j \leq s$. If we choose μ to be a linear functional in the dual of H^n, i.e.

$\lambda = (\lambda_j)_{j=1}^s \in H^{-n}$ for general $n \le 2m$, we get

$$\begin{aligned}
\|\Phi_{l,k}^{i,j} * \lambda_j\|_{2m-n}^2 &= \int\limits_{\mathbb{R}^s} (1+\|\xi\|^2)^{2m-n} |\widehat{\Phi_{l,k}^{i,j} * \lambda_j}(\xi)|^2 \, d\xi \\
&= \int\limits_{\mathbb{R}^s} (1+\|\xi\|^2)^{2m} |\hat{\Phi}_{l,k}^{i,j}(\xi)|^2 (1+\|\xi\|^2)^{-n} |\hat{\lambda}_j(\xi)|^2 \, d\xi \\
&\le \int\limits_{\mathbb{R}^s} (1+\|\xi\|^2)^{2m} |\hat{\Phi}_{l,k}^{i,j}(\xi)|^2 \, d\xi \int\limits_{\mathbb{R}^s} (1+\|\xi\|^2)^{-n} |\hat{\lambda}_j(\xi)|^2 \, d\xi \\
&= \|\Phi_{l,k}^{i,j}\|_{2m}^2 \|\lambda_j\|_{-n}^2,
\end{aligned}$$

and hence $\Phi_{l,k}^{i,j} * \lambda_j \in H^{2m-n}$ and $\|\Phi_{l,k}^{i,j} * \lambda_j\|_{2m-n} \le \|\Phi_{l,k}^{i,j}\|_{2m} \|\lambda_j\|_{-n}$ for all $1 \le i,j \le s$. If we choose $\lambda \in H^{-m}$, i.e. $n = m$, we see that we have $\Phi_{l,k}^{i,j} * \lambda_j \in H^m$ for all $1 \le i,j \le s$, and hence we can apply another $\lambda \in H^{-m}$ to the convolution. Therefore, $(\lambda^* \otimes \lambda, \Phi_{l,k})$ is well defined, and we obtain

$$\begin{aligned}
\int\limits_{\mathbb{R}^s} \lambda(x)^* (\Phi_{l,k} * \lambda)(x) \, dx &= \int\limits_{\mathbb{R}^s} \lambda(x)^* \int\limits_{\mathbb{R}^s} \Phi_{l,k}(x-y)\lambda(y) \, dy \, dx \\
&= \int\limits_{\mathbb{R}^s} \lambda(x)^* \int\limits_{\mathbb{R}^s} \int\limits_{\mathbb{R}^s} e^{i(x-y)\cdot\xi} \, d\mu_{l,k}(\xi) \lambda(y) \, dy \, dx \\
&= \int\limits_{\mathbb{R}^s} \int\limits_{\mathbb{R}^s} (e^{-ix\cdot\xi}\lambda(x))^* \, dx \, d\mu_{l,k}(\xi) \int\limits_{\mathbb{R}^s} \int\limits_{\mathbb{R}^s} (e^{-iy\cdot\xi}\lambda(y))^* \, dy.
\end{aligned}$$

This now yields

$$(\lambda^* \otimes \lambda, \Phi_{l,k}) = \int\limits_{\mathbb{R}^s} \hat{\lambda}(\xi)^* \, d\mu_{l,k}(\xi) \, \hat{\lambda}(\xi) \ge 0 \tag{3.4}$$

for all $\lambda \in H^{-m}$, since $d\mu_{l,k}$ is a positive measure, and (3.4) is equal to zero if and only if λ is equal to zero. This completes the proof. □

We now state some remarks.

Remark 3.3. Instead of divergence-free matrix-valued functions, we are also able to construct classes of functions having other properties, as for example curl-free matrix-valued functions. This applies to both compactly-supported and smooth RBFs. We just have to adjust the measure used in equation (3.2) of Theorem 3.2 to obtain the desired property. Therefore, similar results can be obtained for a large class of functions with certain properties.

Remark 3.4. Equation (3.5) of the previous proof gives us the direct form of $\Phi_{l,k}$ as a byproduct, i.e.

$$\Phi_{l,k}(x) = \{-\Delta I + \nabla\nabla^T\}\,\psi_{l,k}(x) \tag{3.5}$$

for all $x \in \mathbb{R}^s$.

An other important consequence of Theorem 3.2 is stated next.

Lemma 3.5. *The columns of the functions* $\Phi_{l,k}(x) = \{-\Delta I + \nabla\nabla^T\}\,\psi_{l,k}(x)$*, where* $\psi_{l,k}$ *is a scalar-valued Wendland function for* $k \geq 2$*, are divergence-free.*

Proof. Recall that a vector-valued function f is divergence-free if and only if the equation $\nabla \cdot f(x) \equiv 0$ holds for all $x \in \mathbb{R}^s$. Let $\Phi^i_{l,k}(x)$ be an arbitrary column of $\Phi_{l,k}(x)$, for $1 \leq i \leq s$. Then, for $k \geq 2$,

$$\begin{aligned}
\nabla \cdot \Phi^i_{l,k}(x) &= \nabla \cdot \{-\Delta I + \nabla\nabla^T\}_i\,\psi_{l,k}(x) \\
&= \nabla \cdot (-\delta_{ij}\sum_{l=1}^{s}\partial^2_{x(l)} + \partial_{x(i)}\partial_{x(j)})^s_{j=1}\,\psi_{l,k}(x) \\
&= \{-\sum_{l\neq i}\partial^2_{x(l)}\partial_{x(i)} + \sum_{r\neq i}\partial_{x(i)}\partial^2_{x(r)}\}\,\psi_{l,k}(x) \\
&= 0
\end{aligned}$$

for any $\psi_{l,k}(x)$ on $\mathbb{R}^s$ with $k \geq 2$. □

Based on the formula (3.5) we can investigate explicit compactly-supported divergence-free matrix-valued functions. For example, if $k = 2$ and $s = 2$, we choose $l = 4$ by Lemma 3.1, and we obtain the function with C^2 entries

$$\Phi_{4,2}(x) \doteq \{-\Delta I + \nabla\nabla^T\}(1 + \|x\|)_+^6(35\|x\|^2 + 18\|x\| + 3).$$

This can be calculated directly in the following way. A short calculation gives that for general $\psi_{l,k}$ with $r = \|x\|$, $x \in \mathbb{R}^s$, there holds:

$$\Phi_{l,k}(r) \doteq -\frac{s-2}{r}\psi'_{l,k}(r)I - \psi''_{l,k}(r)I + \frac{1}{r}\left(\frac{\psi'_{l,k}(r)}{r}\right)' xx^T.$$

For $s = 2$, this becomes

$$\Phi_{l,k}(r) \doteq -\psi''_{l,k}(r)I + \frac{1}{r}\left(\frac{\psi'_{l,k}(r)}{r}\right)' xx^T,$$

which can be used to calculate $\Phi_{4,2}$ above explicitly.

B. Divergence-free Smooth Functions

For the remainder of this chapter we investigate some divergence-free C^∞ matrix-valued functions as introduced in [22]. We present a direct proof of the strictly positive definiteness of the divergence-free matrix-valued RBFs generated by the Gaussian function $\psi_\alpha(x) := e^{-\alpha\|x\|^2} \in C^\infty$, for $x \in \mathbb{R}^s$ and $\alpha > 0$. Let us define the matrix-valued function

$$\Phi_\alpha(x) = \{-\Delta I + \nabla\nabla^T\}\,\psi_\alpha(x)$$

for $x \in \mathbb{R}^s$. This definition is very similar to the definition of the compactly-supported matrix-valued function in section A, but we use a smooth RBF here instead. We show that Φ_α is divergence free and that the tensor $A_\alpha := (\Phi_\alpha(x_j - x_k))_{j,k=1}^N$ is strictly positive definite for any set $X := \{x_j\}_{j=1}^N \subset \mathbb{R}^2$ of pairwise disjoint points in the case of $s = 2$. A short calculation gives the explicit form

$$\Phi_\alpha(x) = \{(2(s-1)\alpha - 4\alpha^2\|x\|^2)I + 4\alpha^2 x x^T\}\, e^{-\alpha\|x\|^2} \tag{3.6}$$

on $\mathbb{R}^s$, as stated in [22].

Note that Φ_α is divergence free on $\mathbb{R}^s$, by the same argument used as in section A, since we apply the same measure to create the matrix-valued function Φ_α. It now remains to show that the tensor A_α is strictly positive definite.

Proposition 3.6. *The tensor $A_\alpha := (\Phi_\alpha(x_j - x_k))_{j,k=1}^N$ is strictly positive definite, i.e.*

$$c^T A_\alpha c \geq 0 \textit{ for all } c \in \mathbb{R}^2 \times \mathbb{R}^N$$

and

$$c^T A_\alpha c = 0 \textit{ if and only if } c = 0.$$

Proof. We first show that A_α is positive definite, i.e. $c^T A_\alpha c \geq 0$ for all c in $\mathbb{R}^2 \times \mathbb{R}^N$. By definition,

$$\Phi_\alpha(x) = \int_{\mathbb{R}^2} e^{ix\cdot\xi}\{\|\xi\|^2 I - \xi\xi^T\}\, e^{-\|\xi\|^2/4\alpha} \frac{d\xi}{(4\pi\alpha)^{3/2}},$$

since $\hat{\psi}_\alpha(\xi) = (\pi/\alpha)e^{-\|\xi\|^2/(4\alpha)}$. Then,

$$\begin{aligned} c^T A_\alpha c &= \sum_{j,k=1}^N c_j^T \Phi_\alpha(x_j - x_k) c_k \\ &= \sum_{j,k=1}^N c_j^T \left(\int_{\mathbb{R}^2} e^{i(x_j - x_k)\cdot\xi}\{\|\xi\|^2 I - \xi\xi^T\}\, e^{-\|\xi\|^2/4\alpha} \frac{d\xi}{(4\pi\alpha)^{3/2}} \right) c_k, \end{aligned}$$

and therefore,

$$\begin{aligned} c^T A_\alpha c &= \int\limits_{\mathbb{R}^2} \left(\sum_{j=1}^{N} c_j^T e^{ix_j\cdot\xi} \right) \{\|\xi\|^2 I - \xi\xi^T\} \left(\sum_{k=1}^{N} c_k^T e^{ix_k\cdot\xi} \right)^* \\ &\quad \times e^{-\|\xi\|^2/4\alpha} \frac{d\xi}{(4\pi\alpha)^{3/2}}. \end{aligned}$$

By setting $b(\xi) := \sum\limits_{j=1}^{N} c_j^T e^{ix_j\cdot\xi}$ we obtain

$$\begin{aligned} c^T A_\alpha c &= \int\limits_{\mathbb{R}^2} \{\|\xi\|^2 b(\xi) b(\xi)^* - b(\xi)\xi\xi^T b(\xi)^*\} \, e^{-\|\xi\|^2/4\alpha} \frac{d\xi}{(4\pi\alpha)^{3/2}} \\ &= \int\limits_{\mathbb{R}^2} \{\|\xi\|^2 \|b(\xi)\|^2 - \|b(\xi)\xi\|^2\} \, e^{-\|\xi\|^2/4\alpha} \frac{d\xi}{(4\pi\alpha)^{3/2}}. \end{aligned}$$

Applying Cauchy-Schwarz's inequality yields $\|\xi\|^2\|b(\xi)\|^2 - \|b(\xi)\xi\|^2 \geq 0$ and consequently the integral only consists of nonnegative terms. We obtain

$$c^T A_\alpha c = \int\limits_{\mathbb{R}^2} \{\|\xi\|^2 \|b(\xi)\|^2 - \|b(\xi)\xi\|^2\} \, e^{-\|\xi\|^2/4\alpha} \frac{d\xi}{(4\pi\alpha)^{3/2}} \geq 0$$

for all $c \in \mathbb{R}^2 \times \mathbb{R}^{\mathrm{N}}$, which proves the positive definiteness of A_α.

We are left to show that A_α is strictly positive definite, i.e. $c^T A_\alpha c = 0$ if and only if $c = 0$. Note that for all $\xi \in \mathbb{R}^2$, $c^T A_\alpha c = 0$ is equivalent to the statement that $\|\xi\|^2\|b(\xi)\|^2 - \|b(\xi)\xi\|^2 = 0$, which implies that $b(\xi)$ is parallel to ξ. Therefore, there exists a function $k : \mathbb{R}^2 \to \mathbb{R}$ such that $b(\xi) = k(\xi)\xi$, which is equivalent to

$$\sum_{j=1}^{N} c_j^T e^{ix_j\cdot\xi} = k(\xi)\xi. \tag{3.7}$$

Let $c_j = (c_j(1), c_j(2))^T$, $\xi = (\xi(1), \xi(2))^T \in \mathbb{R}^2$ and $\xi^\perp := (-\xi(2), \xi(1))^T$. Then it follows with equation (3.7):

$$
\begin{aligned}
0 &\equiv k(\xi)\xi\xi^{\perp} = \sum_{j=1}^{N} c_j^T e^{ix_j\cdot\xi}(-\xi(2),\xi(1))^T \\
&= \sum_{j=1}^{N}(-c_j(1)\xi(2)+c_j(2)\xi(1))e^{ix_j\cdot\xi}. \qquad (3.8)
\end{aligned}
$$

Since $X = \{x_j\}_{j=1}^N$ is a set of pairwise disjoint points, for an arbitrary but fixed $k \in \{1,\dots,N\}$ there exists a function $f_k : \mathbb{R}^2 \to \mathbb{R}$, $f_k \in C^\infty$, with the following properties:

i) f_k has Fourier transform $\hat{f}_k : \mathbb{R}^2 \to \mathbb{R}$,

ii) $\hat{f}_k$ has local support, i.e. $\hat{f}_k(\cdot - x_k)$ only lives in a small neighborhood of x_k,

iii) $\hat{f}_k$ behaves like a *plateau function*, i.e. $\partial_{x(j)}\hat{f}_k(\cdot - x_k) \equiv 0$ for $j \in \{1,2\}$ and $\partial_{x(l)}\hat{f}_k(\cdot - x_k) \equiv 1$ in a neighborhood of x_k for $l \in \{1,2\}$ with $l \neq j$.

Multiplying (3.8) by $e^{-ix\cdot\xi}f_k(\xi)$ and integrating this expression yields

$$
\begin{aligned}
0 &\equiv \int_{\mathbb{R}^2}\sum_{j=1}^{N}(-c_j(1)\xi(2)+c_j(2)\xi(1))e^{ix_j\cdot\xi}e^{-ix\cdot\xi}f_k(\xi)\,d\xi \\
&= \sum_{j=1}^{N} c_j(1)\int_{\mathbb{R}^2}\xi(2)e^{-i(x-x_j)\cdot\xi}f_k(\xi)\,d\xi + \sum_{j=1}^{N} c_j(2)\int_{\mathbb{R}^2}\xi(1)e^{-i(x-x_j)\cdot\xi}f_k(\xi)\,d\xi.
\end{aligned}
$$

Since

$$\partial_{x(l)}e^{-i(x-x_j)\cdot\xi} = -i\xi(l)e^{-i(x-x_j)\cdot\xi}, \text{ for } l = 1,2,$$

we get

$$0 = -i\sum_{j=1}^{N} c_j(1)\int_{\mathbb{R}^2}\partial_{x(2)}e^{-i(x-x_j)\cdot\xi}f_k(\xi)\,d\xi - i\sum_{j=1}^{N} c_j(2)\int_{\mathbb{R}^2}\partial_{x(1)}e^{-i(x-x_j)\cdot\xi}f_k(\xi)\,d\xi.$$

Using Fourier transform arguments and applying property ii) of $\hat{f}_k$ gives

$$\begin{aligned} 0 &= \sum_{j=1}^{N} c_j(1)\partial_{x(2)}\hat{f}_k(x-x_j) + \sum_{j=1}^{N} c_j(2)\partial_{x(1)}\hat{f}_k(x-x_j) \\ &= c_k(1)\partial_{x(2)}\hat{f}_k(x-x_j) + c_k(2)\partial_{x(1)}\hat{f}_k(x-x_j). \end{aligned}$$

We now apply property ii) of $\hat{f}_k$ twice to obtain the desired result. Firstly, set $\partial_{x(1)}\hat{f}_k(x-x_j) \equiv 0$ in a neighborhood of x_k and let $\partial_{x(2)}\hat{f}_k(x-x_j) \equiv 1$. This results in $c_k(1) = 0$. Secondly, set $\partial_{x(1)}\hat{f}_k(x-x_j) \equiv 1$ in a neighborhood of x_k and let $\partial_{x(2)}\hat{f}_k(x-x_j) \equiv 0$. This leads to $c_k(2) = 0$. Since k was arbitrarily chosen, it follows that $c_k = 0$ for all $k \in \{1, \ldots, N\}$. This completes the proof. $\square$

Remark 3.7. The previous proof can be applied to any C^∞ divergence-free matrix-valued RBF generated by a scalar-valued RBF with positive Fourier transform.

CHAPTER IV

DENSITY THEOREM FOR MATRIX-VALUED RBFS

In this chapter we derive a density result that guarantees that any sufficiently smooth divergence-free function can be approximated arbitrarily closely by a linear combination of divergence-free RBFs. So far, similar results have only been obtained for scalar-valued radial basis functions, see [24, Brown's Theorem]. We give a generalization that applies to the approximation of a large class of vector-valued functions. The result justifies the interpolation approach employing matrix-valued radial basis functions.

A. Preliminaries

We base the approximation problem on the class of matrix-valued RBFs Φ generated by the compactly-supported positive definite Wendland functions discussed in Chapter III. Let

$$\Phi_{l,k}(x) = \{-\Delta I + \nabla\nabla^T\}\psi_{l,k}(x), \tag{4.1}$$

where $\psi_{l,k} \in C^{2k}(\mathbb{R}^s)$ is a Wendland function with $\hat{\psi}_{l,k}(\xi) > 0$ for all $\xi \in \mathbb{R}^s$.

Let us investigate some properties of $\Phi_{l,k}$ and its Fourier transform. Firstly, if $k \geq 2$, then $\psi_{l,k}$ is at least in C^4. This and the compact support of $\psi_{l,k}$ guarantee that $\Phi_{l,k}(x)c$ is in $H^1(\mathbb{R}^s; \mathbb{C}^s)$ for $c \in \mathbb{C}^s$ and $k \geq 2$. Secondly, $\nabla \cdot (\Phi_{l,k}c) \equiv 0$ on $\mathbb{R}^s$ as shown in Lemma 3.5. Finally, as derived in equation (3.5), $\hat{\Phi}_{l,k}(\xi) = \{\|\xi\|^2 I - \xi\xi^T\}\hat{\psi}_{l,k}$ by Fourier transform arguments.

B. Density Theorem and Proof

We now state the density result and conclude the chapter with its proof.

Theorem 4.1. *Let $\Phi_{l,k}$ be given as in (4.1) and define the spaces*

$$\begin{aligned}
(4.2)\quad \mathcal{V} &:= \left\{ \sum_{j=1}^{N} \Phi_{l,k}(\cdot - x_j)c_j \;:\; x_j \in \mathbb{R}^s, c_j \in \mathbb{C}^s \text{ for } 1 \le j \le N, N \in \mathbb{N} \right\}, \\
(4.3)\quad \mathcal{L} &:= \left\{ f \in H^1(\mathbb{R}^s; \mathbb{C}^s) \text{ such that } \nabla \cdot f \equiv 0 \; a.e. \text{on } \mathbb{R}^s \right\}.
\end{aligned}$$

Then $\mathcal{V}$ is dense in $\mathcal{L}$, i.e. any divergence-free vector-valued function $f : \mathbb{R}^s \to \mathbb{C}^s$ in the Sobolev space H^1 can be approximated arbitrarily well by a linear combination of divergence-free matrix-valued RBFs generated by a Wendland function.

Proof. We prove the statement in two steps. First, we show that $\mathcal{L}$ is closed. Then we prove that $\mathcal{V}$ is dense in $\mathcal{L}$.

i) To show: $\mathcal{L}$ is closed, i.e. $\bar{\mathcal{L}} = \mathcal{L}$.

We show that $\mathcal{L}$ is sequentially closed. Let $\{f_n\} \in \mathcal{L}$ be a series of functions such that $f_n \to f \in \bar{\mathcal{L}}$. Then $f_n \in H^1$ and $\nabla \cdot f_n \equiv 0$ $a.e.$ for all n. Certainly, $f \in H^1$, since H^1 is closed. Hence, it remains to show that $f \in \bar{\mathcal{L}}$ is divergence free almost everywhere as well. Now, $f_n \to f$ is equivalent to $\|f_n - f\|_{H^1} \to 0$. However,

$$\begin{aligned}
\|f_n - f\|_{H^1} &\ge \sum_{i=1}^{s} \|\partial_{x(i)}\{f_{n,i} - f_i\}\|_{L^2} \ge \|\sum_{i=1}^{s} \partial_{x(i)}\{f_{n,i} - f_i\}\|_{L^2} \\
&\ge \|\nabla \cdot \{f_n - f\}\|_{L^2} = \|\nabla \cdot f\|_{L^2} \ge 0,
\end{aligned}$$

since $\nabla \cdot f_n \equiv 0$ $a.e.$ on $\mathbb{R}^s$. Here, $f_{n,i}$, f_i are the components of f_n, f, respectively, and we applied the triangle inequality to the L_2-norm. Now, $f_n \to f$ implies that $\nabla \cdot f \equiv 0$ $a.e.$ on $\mathbb{R}^s$, what had to be shown. Hence, $\mathcal{L}$ is closed.

Note that $\nabla \cdot f \equiv 0$ $a.e.$ on $\mathbb{R}^s$ implies that

$$(\nabla \cdot f)^\wedge(\xi) = i\xi^* \hat{f}(\xi) \equiv 0 \; a.e. \text{ on } \mathbb{R}^s. \tag{4.4}$$

ii) To show: $\bar{\mathcal{V}} = \mathcal{L}$, i.e. $\mathcal{V}$ is dense in $\mathcal{L}$.

Assume not. Then $\mathcal{V} \subset \mathcal{L}$ but $\mathcal{V} \neq \mathcal{L}$ and hence, there exists a nonzero element $f \in \mathcal{L} \setminus \mathcal{V}$, such that $f \in \mathcal{V}^\perp$, i.e. $\langle g, f \rangle_{H^1} = 0$ for $g \in \mathcal{V}$. Let $g = \Phi_{l,k}(\cdot - x_j)c_j$ with $x_j \in \mathbb{R}^s$ and $c_j \in \mathbb{C}^s$. Then

$$\begin{aligned}
0 &= \langle f, g \rangle_{H^1} = \int_{\mathbb{R}^s} (1 + \|\xi\|^2) \hat{f}(\xi)^* \hat{g}(\xi)\, d\xi \\
&= \int_{\mathbb{R}^s} (1 + \|\xi\|^2) \hat{f}(\xi)^* [\Phi_{l,k}(\cdot - x_j)c_j]^\wedge(\xi)\, d\xi \\
&= \int_{\mathbb{R}^s} e^{-ix_j \cdot \xi} (1 + \|\xi\|^2) \hat{f}(\xi)^* \{\|\xi\|^2 I - \xi\xi^T\} \hat{\psi}_{l,k}(\xi) c_j\, d\xi \\
&= \int_{\mathbb{R}^s} e^{-ix_j \cdot \xi} (1 + \|\xi\|^2) \Big[\|\xi\|^2 \hat{f}(\xi)^* c_j - \underbrace{\hat{f}(\xi)^* \xi}_{\equiv 0 \; a.e.} \xi^T c_j \Big] \hat{\psi}_{l,k}(\xi)\, d\xi
\end{aligned}$$

by equation (4.4). Hence, we obtain that

$$0 = \Big[(1 + \|\xi\|^2) \|\xi\|^2 \hat{f}(\xi)^* c_j \hat{\psi}_{l,k}(\xi) \Big]^\wedge (x_j)$$

for all $x_j \in \mathbb{R}^s$, $c_j \in \mathbb{C}^s$. Now, this Fourier transform is equal to zero if and only if

$$(1 + \|\xi\|^2) \|\xi\|^2 \hat{f}(\xi)^* c_j \hat{\psi}_{l,k}(\xi) \equiv 0 \; a.e.$$

which is equivalent to $\hat{f}(\xi)^* c_j \equiv 0$ $a.e.$ on $\mathbb{R}^s$. Choosing c_j correctly gives that this holds if and only if $\hat{f}(\xi) \equiv 0$ $a.e.$ on $\mathbb{R}^s$, which is equivalent to $f \equiv 0$ $a.e.$ on $\mathbb{R}^s$. But this is a contradiction to the assumption that f is nonzero. Therefore, we obtain that $\mathcal{V}$ is dense in $\mathcal{L}$.

□

We conclude this chapter with some remarks.

Remark 4.2. Note that in Theorem 4.1 we only assume that the scattered data is given on data sites in $\mathbb{R}^s$, not on a local subset $\Omega \subset \mathbb{R}^s$. The classical density arguments, such as Brown's Theorem [24], or Weierstrass' Theorem are all based on local information. In order to adapt our result to the local case, we would only have to show additionally that $\|f_n - f\|_{H^1(\Omega;\,\mathbb{C}^s)}$ can be extended to the norm on the whole $\mathbb{R}^s$ and that the resulting series is still divergence-free. This might be done using Calderon's Extension Theorem [1] and might be of interest for future work.

Remark 4.3. If we have given a compactly-supported function $f \in \mathcal{L}$ with its support contained in an open set $\Omega' \subset \mathbb{R}^s$, and if $supp(\psi_{l,k}) \subset \Omega'$, then we get a local density statement if we choose Ω such that Ω' is contained properly in Ω. In that case, we can focus on local data sites $x_j \in \Omega$ and Theorem 4.1 yields that f can be locally approximated arbitrarily well by a linear combination of divergence-free matrix-valued RBFs generated by a Wendland function.

CHAPTER V

ERROR ESTIMATES

In this chapter we present rates of approximation concerning the matrix-valued RBF interpolation and approximation problem. Error estimates measure the worst deviation of the interpolant from the function generating the data. We assume that Φ is an $s \times s$ matrix-valued, strictly order-m $\mathcal{S}$-CPD function whose components are in $C^k(\mathrm{I\!R}^s)$, for k even. Set $K = k/2$. We want to obtain error estimates for the interpolants described in Problem 2.4, given that the function $f \in C^K(\mathrm{I\!R}^s)$ generating the data has the form $f = \Phi * \nu + p$, with polynomial $p \in \mathcal{P}_m := \mathrm{I\!P}_m^{s \mapsto s} \cap \mathcal{S}$ and $\nu \in \mathcal{P}_m^{\perp}$, which we define to be the subspace of $\mathcal{E}'_{s,m}(\mathcal{S})$ consisting of s-variant distributions defined on functions in $C^K(\mathrm{I\!R}^s)$. Note that all results from Chapter III apply here as well. There have been only three other works [18, 23, 30] that deal with bounds obtained in a similar way. In [18], the bounds are not explicitly evaluated, and in [30], the bounds are obtained only for the case where the order of the derivative is equal to the order of the polynomial. Narcowich, Ward, and Wendland [23] obtained error estimates for a derivative of any order less than or equal to the order of the polynomial for scalar-valued radial basis functions. We obtain error estimates that hold for vector-valued radial basis functions. The error estimates are of the form

$$\sup_{x \in W(\omega,\delta) \subset \Omega} \|D^\alpha (f - s_f)(x)\|_\infty \leq |f|_\Phi \, C \, h_{\Phi,X}^\beta$$

for some β, where $h_{\Phi,X}$ is the so-called mesh norm and C is a constant independent of f and N. The general strategy used here was sketched in [23], although details differ in important ways.

A. Preliminary Error Estimates

Assume the data is generated by a function $f \in C^K(\mathbb{R}^s)$, where $K = k/2$. Let η be an s-component distribution in $\mathcal{P}_m^{\perp}$ defined on functions in $C^K(\mathbb{R}^s)$. We want to estimate the quantity $|(\eta^*, f - s_f)|$. For example, if we want pointwise estimates, we set $\eta = \mathbf{v}\delta_x$ with $\mathbf{v} \in \mathbb{R}^s$ and bound the quantity $\sup_{\mathbf{v}\in\mathbb{R}^s, \|\mathbf{v}\|=1} |(\mathbf{v}^*\delta_x, f - s_f)|$. By construction, $(\lambda_j^*, f - s_f) = 0$ for $1 \le j \le N$. By Theorem 2.5 there exist c_j, $1 \le j \le N$, such that $\eta|_{\mathcal{P}_m} = \sum_{j=1}^{N} c_j \lambda_j|_{\mathcal{P}_m}$, since we assume

$$\dim\{\text{span}\{\Lambda\} \cap \mathcal{P}_m^{\perp}\} = \dim\ \mathcal{P}_m,$$

which guarantees that polynomials are reproduced by the generalized Hermite interpolation problem. Thus, the residual distribution, $\eta|_{\mathcal{P}_m} - \sum_{j=1}^{N} c_j \lambda_j|_{\mathcal{P}_m}$, is in $\mathcal{P}_m^{\perp}$. Hence, if we set $s_f = \Phi * \lambda + q$, we obtain

$$\begin{aligned}
(\eta^*, f - s_f) &= ((\eta - \sum_{j=1}^{N} c_j\lambda_j)^*, f - s_f) \\
&= ((\eta - \sum_{j=1}^{N} c_j\lambda_j)^*, \Phi * (\nu - \lambda) + (p - q)) \\
&= ((\eta - \sum_{j=1}^{N} c_j\lambda_j)^*, \Phi * (\nu - \lambda)) \\
&= \langle \nu - \lambda, \eta - \sum_{j=1}^{N} c_j\lambda_j \rangle_{\Phi}. \qquad (5.1)
\end{aligned}$$

If $\eta = \lambda \in \text{span}\{\Lambda\} \cap \mathcal{P}_m^{\perp}$, then the equation (5.1) is equivalent to the statement that $(\lambda^*, f - s_f) = \langle \nu - \lambda, \lambda - \sum_{j=1}^{N} c_j\lambda_j \rangle_{\Phi}$ holds . The left hand side yields

$$(\lambda^*, f - s_f) = (\lambda^*, \Phi * (\nu - \lambda)) + \underbrace{(\lambda^*, p - q)}_{0} = \langle \nu - \lambda, \lambda \rangle_{\Phi}.$$

Since $\lambda = \sum_{j=1}^{N} d_j \lambda_j$ for some d_j, $1 \le j \le N$, we obtain

$$(\lambda^*, f - s_f) = \sum_{j=1}^{N} \bar{d}_j \underbrace{(\lambda_j^*, f - s_f)}_{0} = 0.$$

Hence, $\langle \nu - \lambda, \lambda \rangle_\Phi = 0$, which implies that $\|\nu\|_\Phi^2 = \|\nu - \lambda\|_\Phi^2 + \|\lambda\|_\Phi^2$. Thus,

$$\|\lambda\|_\Phi \le \|\nu\|_\Phi \text{ and } \|\nu - \lambda\|_\Phi \le \|\nu\|_\Phi. \tag{5.2}$$

If we apply the Cauchy-Schwarz inequality to equation (5.1) and use (5.2) to bound the term $\|\nu - \lambda\|_\Phi$, we get

$$|(\eta^*, f - s_f)| \le \|\nu\|_\Phi \|\eta - \sum_{j=1}^{N} c_j \lambda_j\|_\Phi, \text{ where } \eta|_{\mathcal{P}_m} = \sum_{j=1}^{N} c_j \lambda_j|_{\mathcal{P}_m}. \tag{5.3}$$

We are now able to state our first error estimate.

Proposition 5.1. *Let Φ be an $s \times s$ matrix-valued, strictly order-m $\mathcal{S}$-CPD function whose components are in $C^k(\mathrm{I\!R}^{\mathrm{s}})$, with k an even integer. Assume that the function f generating the data is in $C^K(\mathrm{I\!R}^{\mathrm{s}})$, $K = k/2$, and has the form $f = \Phi * \nu + p$, with $p \in \mathcal{P}_m = \mathrm{I\!P}_{\mathrm{m}}^{\mathrm{s} \mapsto \mathrm{s}} \cap \mathcal{S}$ and $\nu \in \mathcal{P}_m^\perp$, the subspace of $\mathcal{E}'_{s,m}(\mathcal{S})$ consisting of s-component distributions defined on functions in $C^K(\mathrm{I\!R}^{\mathrm{s}})$. Given an s-component distribution η in $\mathcal{P}_m^\perp$, the interpolation error yields*

$$|(\eta^*, f - s_f)| \le |f|_\Phi P_{\Phi,\Lambda}^\eta, \tag{5.4}$$

where the power function *is defined as*

$$P_{\Phi,\Lambda}^\eta := \min_{\sum_{j=1}^{N} c_j \lambda_j|_{\mathcal{P}_m} = \eta|_{\mathcal{P}_m}} \|\eta - \sum_{j=1}^{N} c_j \lambda_j\|_\Phi. \tag{5.5}$$

Proof. In equation (5.3), replace $\|\nu\|_\Phi$ by the function norm in (2.7), and take the minimum of (5.3) over all c_j's satisfying $\eta|_{\mathcal{P}_m} = \sum_{j=1}^{N} c_j \lambda_j|_{\mathcal{P}_m}$. This gives the result. $\square$

B. The Power Function

In this section we discuss upper bounds on the power function and introduce the idea of norming sets for local polynomial approximation.

1. General Bounds on the Power Function

From Proposition 5.1, we see that $(P^{\eta}_{\Phi,\Lambda})^2$ is the minimum of the quadratic form

$$Q_\Phi(c_1, \ldots, c_N) := ((\eta - \sum_{j=1}^{N} c_j\lambda_j)^* \otimes (\eta - \sum_{j=1}^{N} c_j\lambda_j), \Phi) = \|\eta - \sum_{j=1}^{N} c_j\lambda_j\|^2_\Phi, \tag{5.6}$$

with $\eta|_{\mathcal{P}_m} = \sum_{j=1}^{N} c_j\lambda_j|_{\mathcal{P}_m}$.

Let us investigate the space $\mathrm{I\!P}^{\mathrm{d}\mapsto\mathrm{s}}_{\mathrm{m}}$. If $s = 1$, the space is equal to $\mathrm{I\!P}^{\mathrm{d}}_{\mathrm{m}}$, the space of scalar-valued polynomials of degree less than or equal to $m-1$. The dimension of $\mathrm{I\!P}^{\mathrm{d}}_{\mathrm{m}}$ is $\binom{m-1+d}{d}$. Since each dimension of an s-dimensional polynomials in $\mathrm{I\!P}^{\mathrm{d}\mapsto\mathrm{s}}_{\mathrm{m}}$ can be considered independently, the dimension of $\mathrm{I\!P}^{\mathrm{d}\mapsto\mathrm{s}}_{\mathrm{m}}$ is $s\binom{m-1+d}{d}$. Hence, we can easily determine a basis $\{p_1, \ldots, p_M\}$ for $\mathrm{I\!P}^{\mathrm{d}\mapsto\mathrm{s}}_{\mathrm{m}}$, consisting of $M = s\binom{m-1+d}{d}$ monomials in s dimensions.

The idea is to approximate the $s \times s$ matrix-valued function Φ by a matrix-valued function $\Phi_{\mathcal{P}_m}$ whose components are in $\mathcal{P}_m$, and to estimate Q_Φ by bounding $Q_{\Phi-\Phi_{\mathcal{P}_m}}$. Note that $\mathcal{P}_m$ might only consist of constant or linear polynomials. Hence, the space $\mathcal{P}_m$ might seem rather small. The question naturally arises if $\mathcal{P}_m$ is big enough to contain at least one polynomial that approximates Φ as desired and that is in the admissible space $\mathcal{S}$. The following two propositions answer the question.

Proposition 5.2. *Let* $\mathcal{P}_0 := \mathbb{P}_0 \cap \mathcal{S}$ *be a space fulfilling* $\mathbb{P}_0 \supseteq \mathbb{P}_{\mathrm{m}}^{\mathrm{s}\mapsto\mathrm{s}}$*, for given* $\mathcal{P}_m = \mathbb{P}_{\mathrm{m}}^{\mathrm{s}\mapsto\mathrm{s}} \cap \mathcal{S}$*. Then* Φ *being a strictly* $\mathcal{S}$*-CPD function with regard to* $\mathcal{P}_m$ *implies that* Φ *is a strictly* $\mathcal{S}$*-CPD function with regard to* $\mathcal{P}_0$*.*

Proof. First note that if we enlarge $\mathcal{P}_m$ to a bigger space $\mathcal{P}_0 = \mathbb{P}_0 \cap \mathcal{S}$ with $\mathbb{P}_0 \supseteq \mathbb{P}_{\mathrm{m}}^{\mathrm{s}\mapsto\mathrm{s}}$, which contains more functions that are in the admissible space $\mathcal{S}$, we get $\mathcal{E}_0' \subseteq \mathcal{E}_{s,m}'$, where

$$\mathcal{E}_0' := \{\mu \in \mathcal{E}_s' : (\mu^*, p) = 0 \text{ for all } p \in \mathcal{P}_0\},$$

and $\mathcal{E}_{s,m}'$ is defined as in equation (2.2). But this yields that Φ being a strictly $\mathcal{S}$-CPD function with regard to $\mathcal{P}_m$ implies that Φ is a strictly $\mathcal{S}$-CPD function with regard to $\mathcal{P}_0$. $\square$

Proposition 5.3. *There always exists a polynomial* $\Phi_{\mathcal{P}_m}$ *which approximates* Φ *as desired and which is still in the admissible space* $\mathcal{S}$*.*

Proof. First, note that Proposition 5.2 guarantees the existence of a polynomial $\Phi_{\mathcal{P}_m}$ which is arbitrarily close to the original function Φ in each of its components. We now have to prove that the columns of $\Phi_{\mathcal{P}_m}$ are in the admissible space $\mathcal{S}$. Let $\mathcal{S} = \mathcal{S}_B$ as introduced in Definition 2.2, and let each component $\Phi_{\mathcal{P}_m \rho,\sigma}$ of $\Phi_{\mathcal{P}_m}$ be the Taylor polynomial of degree $k-1$ at a neighborhood of the origin for the component $\Phi_{\rho,\sigma}$ of Φ, where $1 \leq \rho, \sigma \leq s$. Since the columns of Φ are in the admissible space $\mathcal{S}_B$, we have to check if this holds for $\Phi_{\mathcal{P}_m}$ as well. Fortunately it does, by the following argument. First, observe that since the columns Φ_i of Φ are in the admissible space $\mathcal{S}_B$, they fulfill $B_j(\nabla)^*\Phi_i \equiv 0$ for all $1 \leq j \leq \nu$ and all $1 \leq i \leq s$. Hence, if we show that this also holds for the columns of $\Phi_{\mathcal{P}_m}$, we are done. Now,

$$\Phi_{\mathcal{P}_m \rho,\sigma}(x) = \sum_{|\beta|<k} D^\beta \Phi_{\rho,\sigma}(0)\, \frac{x^\beta}{\beta!}.$$

Let us investigate $\partial_{x(i)}\Phi_{\mathcal{P}_m\rho,\sigma}$. We have

$$\begin{aligned}\partial_{x(i)}\Phi_{\mathcal{P}_m\rho,\sigma}(x) &= \partial_{x(i)}\sum_{|\beta|<k} D^{\beta}\Phi_{\rho,\sigma}(0)\,\frac{x^{\beta}}{\beta!}\\ &= \sum_{|\beta|<k} D^{\beta_1,\ldots,\beta_n}\Phi_{\rho,\sigma}(0)\,\partial_{x(i)}\frac{x_1^{\beta_1}\cdots x_n^{\beta_n}}{\beta_1!\cdots\beta_n!}\\ &= \sum_{|\beta|<k-1,\beta_i\geq 1} D^{\beta_1,\ldots,\beta_n}\Phi_{\rho,\sigma}(0)\,\frac{x_1^{\beta_1}\cdots x_i^{\beta_i-1}\cdots x_n^{\beta_n}}{\beta_1!\cdots(\beta_i-1)!\cdots\beta_n!},\end{aligned}$$

since the terms with $\beta_i = 0$ become zero. But since β_i is at least 1, we can pull one $\partial_{x(i)}$ out of the coefficients of the Taylor series. We get

$$\begin{aligned}\partial_{x(i)}\Phi_{\mathcal{P}_m\rho,\sigma}(x) &= \sum_{|\beta|<k-1,\beta_i\geq 1}((\partial_{x(i)}D^{\beta_1,\ldots,\beta_i-1,\ldots,\beta_n})\Phi_{\rho,\sigma})(0)\,\frac{x_1^{\beta_1}\cdots x_i^{\beta_i-1}\cdots x_n^{\beta_n}}{\beta_1!\cdots(\beta_i-1)!\cdots\beta_n!}\\ &= \sum_{|\tilde\beta|<k-1}((\partial_{x(i)}D^{\tilde\beta})\Phi_{\rho,\sigma})(0)\,\frac{x^{\tilde\beta}}{\tilde\beta!} = \sum_{|\tilde\beta|<k-1}(D^{\tilde\beta}(\partial_{x(i)}\Phi_{\rho,\sigma}))(0)\,\frac{x^{\tilde\beta}}{\tilde\beta!},\end{aligned}$$

where $\tilde\beta = \beta|_{\beta_i\to\beta_i-1}$. Hence, partial derivatives turn out to work on the coefficients. For general $D^{\alpha}\Phi_{\rho,\sigma}$, the same idea applies, since any $\partial_{x(i)}\partial_{x(j)}$ can be written as $\partial_{x(i)}(\partial_{x(j)})$, and are applied one at a time. Therefore, we obtain

$$\begin{aligned}B_j(\nabla)^*\Phi_{\mathcal{P}_m,i} &= B_j(\nabla)^*\left(\sum_{|\beta|<k} D^{\beta}\Phi_i(0)\,\frac{x^{\beta}}{\beta!}\right)\\ &= \sum_{|\tilde\beta|<\tilde k}(D^{\beta}(\underbrace{B_j(\nabla)^*\Phi_i}_{0}))(0)\,\frac{x^{\tilde\beta}}{\tilde\beta!} = 0,\end{aligned}$$

for any column $\Phi_{\mathcal{P}_m,i}$ of $\Phi_{\mathcal{P}_m}$, where $1\leq i\leq s$, since the derivatives D^{β} and the B_j's are commutative, and $\tilde\beta$, $\tilde k$ are obtained in the above described matter. $\square$

Hence, the space $\mathcal{P}_m$ can always be enlarged to contain functions that approximate Φ as desired. Also, note that $\Phi-\Phi_{\mathcal{P}_m}$ is at least $C^K(\mathbb{R}^s)$ component-wise. We need the following proposition.

Proposition 5.4. *Let $\Phi_{\mathcal{P}_m}$ be an $s \times s$ matrix-valued function whose components are in $\mathcal{P}_m$, and let the c_j's satisfy the constraint*

$$\eta|_{\mathcal{P}_m} = \sum_{j=1}^{N} c_j \lambda_j|_{\mathcal{P}_m}.$$

Then $Q_\Phi(c_1, \ldots, c_N) = Q_{\Phi - \Phi_{\mathcal{P}_m}}(c_1, \ldots, c_N)$ and

$$(P^{\eta}_{\Phi,\Lambda})^2 \leq Q_{\Phi-\Phi_{\mathcal{P}_m}}(c_1, \ldots, c_N).$$

Proof. Let $\mu = \eta - \sum\limits_{j=1}^{N} c_j \lambda_j$. Then $\mu \in \mathcal{P}_m^{\perp}$ by the nature of the constraint. Hence, $Q_\Phi = (\mu^* \otimes \mu, \Phi)$. If $\{p_1, \ldots, p_M\}$ is a basis for $\mathcal{P}_m$, we write $\Phi_{\mathcal{P}_m} = \sum_{j,k=1}^{M} b_{j,k} p_j \otimes p_k^*$, where $b_{j,k}$ and $p_j \otimes p_k^*$ are matrices for all j, k. Thus,

$$\begin{aligned}
Q_{\Phi-\Phi_{\mathcal{P}_m}} &= (\mu^* \otimes \mu, \Phi) - (\mu^* \otimes \mu, \Phi_{\mathcal{P}_m}) \\
&= (\mu^* \otimes \mu, \Phi) - \sum_{j,k=1}^{M} \underbrace{(\mu^* \otimes \mu, b_{j,k} p_j \otimes p_k^*)}_{0} \\
&= (\mu^* \otimes \mu, \Phi) \\
&= Q_\Phi.
\end{aligned}$$

By definition, $(P^{\eta}_{\Phi,\Lambda})^2$ is the minimum of Q_Φ, which completes the proof. $\square$

In [22, Lemma 2.2] it was shown that, if $\Phi(x)^* = \Phi(-x)$, then Φ is a conjugate symmetric matrix-valued function. We now require that Φ fulfill this assumption. Then a short calculation gives that $\Phi_{\mathcal{P}_m}$ is conjugate symmetric as well, based on its definition. We obtain the following expression for the power function:

$$\begin{aligned}
Q_{\Phi-\Phi_{\mathcal{P}_m}} &= (\eta^* \otimes \eta, \Phi - \Phi_{\mathcal{P}_m}) - 2\Re\Big\{ \sum_j c_j (\eta^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m}) \Big\} \\
&\quad + \sum_{j,k=1}^{N} c_j c_k^* (\lambda_k^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m}).
\end{aligned}$$

Hence, the quadratic form $Q_{\Phi-\Phi_{\mathcal{P}_m}}$ can be bounded above as follows:

$$
\begin{aligned}
|Q_{\Phi-\Phi_{\mathcal{P}_m}}| &\leq \underbrace{|(\eta^* \otimes \eta, \Phi - \Phi_{\mathcal{P}_m})|}_{=:\Delta_0} + |2\Re\Big\{\sum_j c_j(\eta^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m})\Big\}| \\
&\quad + |\sum_{j,k=1}^{N} c_j c_k^* (\lambda_k^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m})| \\
&\leq \Delta_0 + 2\|c\|_1 \underbrace{\max_j |(\eta^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m})|}_{=:\Delta_1} + \|c\|_1^2 \underbrace{\max_{j,k} |(\lambda_k^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m})|}_{=:\Delta_2} \\
&= \Delta_0 + 2\|c\|_1\Delta_1 + \|c\|_1^2\Delta_2. \qquad (5.7)
\end{aligned}
$$

If we combine this upper bound of the quadratic form $Q_{\Phi-\Phi_{\mathcal{P}_m}}$ with Proposition 5.4, we obtain the following result for the power function.

Theorem 5.5. *Let $\Phi_{\mathcal{P}_m}$ be any conjugate symmetric matrix-valued function whose components are in $\mathcal{P}_m$. For any $c = (c_j)_{j=1}^N$ satisfying the constraint*

$$\eta|_{\mathcal{P}_m} = \sum_{j=1}^{N} c_j \lambda_j|_{\mathcal{P}_m}$$

we have the following upper bound on the power function:

$$(P_{\Phi,\Lambda}^{\eta})^2 \leq \Delta_0 + 2\|c\|_1\Delta_1 + \|c\|_1^2\Delta_2, \qquad (5.8)$$

where

$$
\begin{aligned}
\Delta_0 &:= |(\eta^* \otimes \eta, \Phi - \Phi_{\mathcal{P}_m})|, \\
\Delta_1 &:= \max_j |(\eta^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m})|, \qquad (5.9) \\
\Delta_2 &:= \max_{j,k} |(\lambda_k^* \otimes \lambda_j, \Phi - \Phi_{\mathcal{P}_m})|.
\end{aligned}
$$

2. Norming Sets

The goal is now to estimate the Δ_j's obtained in Theorem 5.5 and to obtain bounds on $\|c\|_1$ that reflect the trade-off between distributions in Λ and η. In order to obtain upper bounds on the norm of c, we apply the method of *norming sets.* Jetter, Stöckler, and Ward give a good description of the method of norming sets in [13, 14]. This was central to [23].

The main idea of norming sets is to find a set $X = \{x_1, \ldots, x_n\} \subset \Omega$ of distinct centers, where Ω is a compact subset of $\mathbb{R}^{\mathrm{s}}$, such that for a given polynomial $p \in \mathcal{P}_m$ with $\|p\| = 1$, we have $\|p|_X\| \geq 1/2$. Therefore, norming sets are used to determine a set of centers X such that the operator $T : \mathcal{P}_m \to \mathbb{R}^N$ given by $T(p) = p|_X$ is injective with $\|T^{-1}\| \leq 2$.

Definition 5.6. Let V be a normed vector space with dual space V^*. Given the subset $Z \subseteq V^*$, then Z is a *norming set for* V if there exists $c > 0$ such that

$$\sup_{z \in Z, \|z\|=1} |z(v)| \geq c\|v\| \text{ for all } v \in V. \tag{5.10}$$

Let $T : V \to T(V) \subseteq \mathbb{C}^{|Z|}$ defined by $T(v) = (z(v))_{z \in Z}$ denote the *sampling operator.*

The norm of the inverse sampling operator is given by

$$\|T^{-1}\| := \sup_{v \in V, v \neq 0} \frac{\|v\|_V}{\|T(v)\|_\infty}. \tag{5.11}$$

Proposition 5.7. *Let Z be a norming set for V with T being the corresponding sampling operator. If $\psi \in V^*$, then there exists $a = \{a_z\}_{z \in Z} \in \mathbb{C}^{|Z|^*}$ depending only on ψ such that for all $v \in V$,*

$$\psi(v) = \sum_{z \in Z} a_z z(v) \text{ and } \|a\|_1 \leq \|\psi\|_{V^*} \|T^{-1}\|. \tag{5.12}$$

To apply this general statement to the power function on a compact domain $\Omega \subset \mathbb{R}^{\mathrm{s}}$ and at given centers $X = \{x_j\}_{j=1}^N$, set $\mathcal{P}_m = \mathbb{P}_{\mathrm{m}}^{\mathrm{s} \mapsto \mathrm{s}} \cap \mathcal{S}$. We take $\Lambda = \{\lambda_j\}_{j=1}^N$ as a set of distributions. We choose V to be $\mathcal{P}_m$, and define the set Z to consist of $(\lambda_j^*|_{\mathcal{P}_m}, \cdot)$, for $1 \leq j \leq N$. Hence, the sampling operator is given by $T(p)_j = (\lambda_j^*, p)$

for $p \in \mathcal{P}_m$. Now, Z is a norming set for V on X, Ω, since $T(p) = 0$ implies that $p = 0$ by Theorem 2.5. This yields that T is injective. If we take $\psi = (\eta^*|_{\mathcal{P}_m}, \cdot)$ in Proposition 5.7, we obtain the following result that gives us a norming set on X, Ω.

Corollary 5.8. *There exist coefficients $c = \{c_j\}_{j=1}^N$ such that for all $p \in \mathcal{P}_m$,*

$$\eta(p) = \sum_{j=1}^{N} c_j \lambda_j(p) \ \textit{and} \ \|c\|_1 \leq \|\eta^*|_{\mathcal{P}_m}\|_{\mathcal{P}_m^*} \|T^{-1}\|,$$

where $\|T^{-1}\|$ is defined in (5.11), and the norm $\|\cdot\|_{\mathcal{P}_m^*}$ depends on that for $\mathcal{P}_m$.

C. Error Estimates on $\mathrm{I\!R}^{\mathrm{s}}$

In this section we want to establish error estimates based on upper bounds for the power function $P_{\Phi,\Lambda}^{\eta}$ for a matrix-valued, strictly order-m $SCPD$ function Φ on a compact domain $\Omega \subset \mathrm{I\!R}^{\mathrm{s}}$. We assume that Φ is in $C_\nu^k(\mathrm{I\!R}^{\mathrm{s}})$, where k is an even integer, i.e. Φ has k derivatives that are Hölder continuous at the origin, with Hölder exponent $0 < \nu \leq 1$.

The spaces $C_\nu^k(\mathrm{I\!R}^{\mathrm{s}})$ are Banach spaces analogous in a certain sense to the spaces $H^k(\mathrm{I\!R}^{\mathrm{s}})$ used to obtain positive definiteness results in Chapter III. In the C_ν^k spaces, weak differentiability is replaced by continuous differentiability and L_2-integrability is replaced by Hölder continuity with exponent ν. Also, it can be shown that for certain choices of k, k', and ν, the space $H^k(\mathrm{I\!R}^{\mathrm{s}})$ can be embedded continuously, or even compactly, in $C_\nu^{k'}(\mathrm{I\!R}^{\mathrm{s}})$, see [9, Theorem 7.26] and [1, Theorem 2.2].

The set Λ comprises vector-valued Dirac δ functions $\lambda_j = v_j \delta_{x_j}$, with $v_j \in \mathrm{I\!R}^{\mathrm{s}}$, for $1 \leq j \leq N$, based on a finite set $X = \{x_1, \ldots, x_n\}$ of distinct points in $\mathrm{I\!R}^{\mathrm{s}}$, such that $X \subset \Omega$. Let the distribution η be given by $\eta = (-1)^{|\alpha|}\, v\, D^\alpha \delta_x$, where x belongs to Ω, $v \in \mathrm{I\!R}^{\mathrm{s}}$, and $|\alpha| \leq k/2$. Let $\mathcal{P}_m$ be $\mathrm{I\!P}_{\mathrm{m}}^{\mathrm{s} \mapsto \mathrm{s}} \cap \mathcal{S}$, the intersection of the admissible

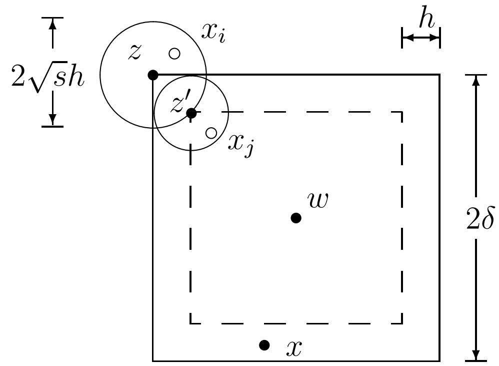

FIGURE 2. The cube $W(w, \delta)$, with mesh norm $h = h_{X,\Omega}$ and space dimension s

space $\mathcal{S}$ and $\mathbb{P}_{\mathrm{m}}^{\mathrm{s}\mapsto\mathrm{s}}$. The *mesh norm*, or Hausdorff distance, for Ω is defined to be

$$h_{X,\Omega}; = \sup_{y\in\Omega} \min_{x_j\in X} \|y - x_j\|_2. \tag{5.13}$$

The estimates we obtain are given in terms of the mesh norm, and hold uniformly for any sub-domain of Ω that can be covered by cubes.

1. Estimates on a Cube

Our objective is to obtain upper bounds for the power function $P^{\eta}_{\Phi,\Lambda}$, where x, related to η, is contained in $W(w, \delta) := \{x \in \mathbb{R}^{\mathrm{s}} : \|x - w\|_\infty \leq \delta\}$, a closed cube in Ω with side length 2δ and center w, see Figure 2. The figure is taken from [23]. We assume for the remainder of the section that $\delta/h_{\Omega,X} > 1$, which guarantees that $Y := X \cap W$ is non-empty. Let us define the *mesh norm for W with respect to Y* as

$$h_{Y,W} := \sup_{z\in W} \min_{x_j\in Y} \|z - x_j\|_2.$$

There are two possibilities, $h := h_{X,\Omega}$ can be either larger than $h_{Y,W}$, or h can be smaller than $h_{Y,W}$. The latter situation is illustrated in Figure 2. Narcowich, Ward, and Wendland showed in [23] that $h_{Y,W} \leq (1+\sqrt{s})h_{X,\Omega}$, using the following argument: Since $h < h_{Y,W}$, the closest point $x_i \in X$ to the corner point z is outside of $W(w,\delta)$. Hence, $x_i \notin Y = X \cap W$. On the other hand, for the point z' on the corner of the cube $W(m, \delta - h)$, there is an element $x_j \in Y$ such that $\|z' - x_j\|_2 \leq h$, since z' has a distance of at least h to the boundary of the outer cube $W(m,\delta)$. Hence, the distance from z to the closest point in Y is bounded above by $\sqrt{s}h + h$. Since this is the maximum distance a point in $W(w,\delta)$ can have to Y, we obtain the above result, i.e.

$$h_{Y,W} \leq (1+\sqrt{s})h_{X,\Omega}. \tag{5.14}$$

We need this result later to relate the mesh norm for the cube to the mesh norm on Ω in our error estimates. We now look at norming sets on the cube. Assume that we have given the cube $W(w,\delta)$, the subset Λ_Y of vector-valued Dirac δ-functions Λ at points in Y, and a polynomial $p \in \mathcal{P}_m$, such that $\|p\|_{\infty,W} = 1$. Let $\mathcal{P}_m = \mathrm{IP}_{\mathrm{m}}^{\mathrm{s}\mapsto\mathrm{s}} \cap \mathcal{S}$ be equipped with the norm

$$\|p\|_{\infty,W} := \sup_{x\in W} \max_{1\leq i\leq s} |p_i(x)|,$$

with p_i being the i^{th} component of p, for $1 \leq i \leq s$. We want to find a norming set $\Lambda_Y \subset \Lambda$ for $\mathcal{P}_m$ such that $\|p|_Y\|_{\infty,W} \geq 1/2$.

Recall that Markov's inequality

$$\|p'\|_{\infty,[a,b]} \leq \frac{2(m-1)^2}{b-a}\|p\|_{\infty,[a,b]} \tag{5.15}$$

holds for $p \in \mathrm{IP}_{\mathrm{m}}^1$ on $[a,b] \in \mathrm{IR}$. If we now extend the equality (5.15) to a polynomial

$p \in \mathbb{P}_{\mathrm{m}}^{\mathrm{s}}$ and let $x \in W = W(m, \delta)$, we get as shown in [23]

$$\|D^\alpha p\|_{\infty,W} \le \left(\frac{(m-1)^2}{\delta}\right)^{|\alpha|} \|p\|_{\infty,W}. \tag{5.16}$$

Note that $\eta = (-1)^{|\alpha|}\, v\, D^\alpha \delta_x$ yields

$$\langle \eta, p \rangle = v^*\, D^\alpha p(x) = \sum_{j=1}^{s} \bar{v}_j (D^\alpha p_j(x)) \le \|v\|_1 \sup_{1 \le i \le s} |D^\alpha p_j(x)|,$$

and hence applying Markov's inequality (5.16) to the upper bound of $\|\eta|_{\mathcal{P}_m}\|_{\mathcal{P}_m^*}$ gives

$$\|\eta|_{\mathcal{P}_m}\|_{\mathcal{P}_m^*} \le \|v\|_1 \left(\frac{(m-1)^2}{\delta}\right)^{|\alpha|}. \tag{5.17}$$

We now are able to prove the following result which gives a norming set on $\mathcal{P}_m$:

Lemma 5.9. *Let $W = W(w, \delta)$ be a cube of side length 2δ and center w, contained in the compact set $\Omega \subset \mathbb{R}^{\mathrm{s}}$, and let $\mathcal{P}_m = \mathbb{P}_{\mathrm{m}}^{\mathrm{s} \mapsto \mathrm{s}} \cap \mathcal{S}$. If the mesh norm on the cube, $h_{Y,W}$, satisfies the condition*

$$h_{Y,W} \le \frac{\delta}{2\sqrt{s}(m-1)^2}, \tag{5.18}$$

then the sampling operator $T : \mathcal{P}_m \to \mathbb{R}^{\mathrm{N}}$ given by $T(p) = p|_Y$ is injective, with $\|T^{-1}\| \le 2$, and Λ_Y is a norming set for $\mathcal{P}_m$.

Proof. Assume that $\|p\|_{\infty,W} = 1$. Since W is a compact subset of Ω, there exists a $z \in W$ and an $l \in \{1, \dots, s\}$ such that $\|p\|_{\infty,W} = |p_l(z)|$. Let y be the closest point in Y to z, i.e. $\|z - y\|_2 \le h_{Y,W}$. Then

$$|p_l(y)| \ge |p_l(z)| - |p_l(z) - p_l(y)| = \|p\|_{\infty,W} - |p_l(z) - p_l(y)|.$$

By applying the mean value theorem to the last term, we obtain the following lower estimate for some $\xi \in \mathbb{R}^s$:

$$|p_l(y)| \geq \|p\|_{\infty,W} - \left| \sum_{j=1}^{s} \frac{\partial p_l}{\partial x(j)}(\xi)(z(j) - y(j)) \right|.$$

Applying Cauchy-Schwarz's inequality and Markov's inequality (5.16) for p_l with $|\alpha| = 1$ yields

$$\begin{aligned} |p_l(y)| &\geq \|p\|_{\infty,W} - \|\nabla p_l(\xi)\|_2 \|z - y\|_2 \\ &\geq \|p\|_{\infty,W} - \frac{\sqrt{s}l^2}{\delta} \|p_l\|_{\infty,W} \|z - y\|_2 \\ &\geq \|p\|_{\infty,W} - \frac{\sqrt{s}l^2}{\delta} \|p\|_{\infty,W} \|z - y\|_2. \end{aligned}$$

If we now use $\|z - y\|_2 \leq h_{Y,W}$ and the condition $h_{Y,W} \leq \delta/[2\sqrt{s}(m-1)^2]$, we obtain that $|p_l(y)| \geq 1/2 \|p\|_{\infty,W}$. Since $\|p|_Y\|_{\infty,W} \geq |p_l(y)|$, we get the estimate $\|p|_Y\|_{\infty,W} \geq 1/2 \|p\|_{\infty,W}$. But this is equivalent to the operator T being injective, with

$$\|T^{-1}\| = \sup_{p \in \mathcal{P}_m, p \neq 0} \frac{\|p\|_{\mathcal{P}_m}}{\|Tp\|_{\infty,W}} \leq 2.$$

□

Combining Lemma (5.9), inequality (5.17), and Corollary 5.8 yields the following result with notation as in Lemma 5.9.

Lemma 5.10. *For every $x \in W = W(w, \delta)$, there exist coefficients $c = \{c_j\}_{j=1}^N$ such that for all $p \in \mathcal{P}_m$,*

$$\langle \eta, p \rangle = v^* D^\alpha p(x) = \sum_{x_j \in Y} c_j \bar{v}_j p(x_j), \text{ with } \|c\|_1 \leq 2\, \|v\|_1 \left(\frac{(m-1)^2}{\delta} \right)^{|\alpha|},$$

provided $h_{Y,W} \leq \delta/[2\sqrt{s}(m-1)^2]$.

After having obtained an upper bound for $\|c\|_1$, we are left to specify bounds on Δ_0, Δ_1, and Δ_2. Those are obtained in the proof of the following theorem that combines all results gathered in this section to get an upper bound on the power function.

Theorem 5.11. *Let $\Phi \in C_\nu^k(\mathbb{R}^s)$ be a matrix-valued, order-m conditionally positive definite function, and, for $x \in W = W(w,\delta)$, let $\eta = (-1)^{|\alpha|} v D^\alpha \delta_x$, with $|\alpha| \le k/2$. If the mesh norm, $h_{Y,W}$, satisfies $h_{Y,W} \le \delta/[2\sqrt{s}(m-1)^2]$, then the power function P^η_{Φ,Λ_Y} possesses the upper bound*

$$
\begin{aligned}
(P^\eta_{\Phi,\Lambda_Y})^2 \quad &\le \quad \left(\frac{s^{(k-|\alpha|)/2}(2\sqrt{s}(m-1)^2)^{|\alpha|}}{(k-|\alpha|)!} + \frac{s^{k/2}(2\sqrt{s}(m-1)^2)^{2|\alpha|}}{k!} \max_{x_j \in Y} \|v_j\|_1 \right) \\
&\qquad \times 4\, M^\Phi_{k,\nu}\, \|v\|_1^2 \max_{x_j\in Y} \|v_j\|_1 \, (2\sqrt{s}\delta)^{k+\nu-2|\alpha|}, \qquad (5.19)
\end{aligned}
$$

where

$$
M^\Phi_{k,\nu} := \max_{1\le\rho,\sigma\le s, |\beta|=k} \|D^\beta \Phi_{\rho,\sigma}\|_{C_\nu}. \qquad (5.20)
$$

Proof. We use Theorem 5.5. Therefore, we are left to find bounds for Δ_0, Δ_1, and Δ_2 defined in equation (5.9). Let us get some upper bounds for the derivatives of a polynomial p first. Define

$$
\|D^\alpha p\|_{\infty,W} := \sup_{x\in W} \max_{1\le j\le s} |D^\alpha p_j(x)|.
$$

Since W is compact, we can find l and $z \in W$ such that $\|D^\alpha p\|_{\infty,W} = |D^\alpha p_l(z)|$. Since $|D^\alpha p_l(z)| \le [(m-1)^2/\delta]^{|\alpha|} \, \|p_l\|_{\infty,W} \le [(m-1)^2/\delta]^{|\alpha|} \, \|p\|_{\infty,W}$ by inequality (5.16), we obtain

$$
\|D^\alpha p\|_{\infty,W} \le \left(\frac{(m-1)^2}{\delta} \right)^{|\alpha|} \|p\|_{\infty,W} \text{ for } p \in \mathbb{P}^s_m \cap \mathcal{S}. \qquad (5.21)
$$

Let $\phi_{\mathcal{P}_m}$ be the Taylor polynomial of degree $k-1$ for a scalar-valued function $\phi \in C^k(\mathbb{R}^s)$, i.e. $\phi_{\mathcal{P}_m}(t) = \sum_{|\beta|<k} D^\beta \phi(0) t^\beta / \beta!$. From the remainder in Taylor's Theorem applied to $D^\gamma \phi$ with $|\gamma| \le k$, Narcowich, Ward, and Wendland obtained in

[23] the inequality

$$|D^\gamma \phi(t) - D^\gamma \phi_{\mathcal{P}_m}(t)| \leq \frac{s^{(k-|\gamma|)/2}\tilde{M}^{\phi}_{k,\nu}}{(k-|\gamma|)!}\|t\|_2^{k+\nu-|\gamma|}, \tag{5.22}$$

for t in a sufficiently small neighborhood of 0 and $\tilde{M}^{\phi}_{k,\nu} := \max_{|\beta|=k}\|D^\beta \phi\|_{C_\nu}$.

Since $\eta = (-1)^{|\alpha|} v D^\alpha \delta_x$, where $|\alpha| \leq k/2$, we get

$$\begin{aligned}
\Delta_0 &= |v^*(D^{2\alpha}\Phi(0) - D^{2\alpha}\Phi_{\mathcal{P}_m}(0))v| \\
&= \Big| \sum_{\rho,\sigma=1}^{s} \bar{v}(\rho)(D^{2\alpha}\Phi_{\rho,\sigma}(0) - D^{2\alpha}\Phi_{\mathcal{P}_m\rho,\sigma}(0))v(\sigma)\Big| \\
&\leq \|v\|_1^2 \max_{1\leq\rho,\sigma\leq s}|D^{2\alpha}\Phi_{\rho,\sigma}(0) - D^{2\alpha}\Phi_{\mathcal{P}_m\rho,\sigma}(0)|.
\end{aligned}$$

Applying inequality (5.22) with $\phi = \Phi_{\rho,\sigma}$, $\gamma = 2\alpha$, and $t = 0$ now yields that

$$\Delta_0 = 0. \tag{5.23}$$

In order to get upper bounds for Δ_1 and Δ_2, we first define $M^{\Phi}_{k,\nu}$ as the quantity

$$M^{\Phi}_{k,\nu} := \max_{1\leq\rho,\sigma\leq s, |\beta|=k}\|D^\beta \Phi_{\rho,\sigma}\|_{C_\nu}. \tag{5.24}$$

Now we obtain the following chain of inequalities for Δ_1:

$$\begin{aligned}
\Delta_1 &= \max_{x_j\in Y}|v^*(D^\alpha\Phi(x-x_j) - D^\alpha\Phi_{\mathcal{P}_m}(x-x_j))v_j| \\
&= \max_{x_j\in Y}\Big| \sum_{\rho,\sigma=1}^{s} \bar{v}(\rho)(D^\alpha\Phi_{\rho,\sigma}(x-x_j) - D^\alpha\Phi_{\mathcal{P}_m\rho,\sigma}(x-x_j))v_j(\sigma)\Big| \\
&\leq \|v\|_1 \max_{x_j\in Y}\Big\{\|v_j\|_1 \max_{1\leq\rho,\sigma\leq s}|D^\alpha\Phi_{\rho,\sigma}(x-x_j) - D^\alpha\Phi_{\mathcal{P}_m\rho,\sigma}(x-x_j)|\Big\} \\
&\leq \|v\|_1 \max_{x_j\in Y}\Big\{\|v_j\|_1 \frac{s^{(k-|\alpha|)/2}}{(k-|\alpha|)!} M^{\Phi}_{k,\nu}\|x-x_j\|_2^{k+\nu-|\alpha|}\Big\},
\end{aligned}$$

where we applied inequality (5.22) with $\gamma = \alpha$ in the last step. Since x and x_j are both points in the cube W, we have $\|x - x_j\|_2 \leq 2\sqrt{s}\delta$. This yields

$$\Delta_1 \leq \|v\|_1 \max_{x_j \in Y} \|v_j\|_1 \frac{s^{(k-|\alpha|)/2}}{(k-|\alpha|)!} M^{\Phi}_{k,\nu} (2\sqrt{s}\delta)^{k+\nu-|\alpha|}. \tag{5.25}$$

Similarly, by setting $\gamma = 0$ in inequality (5.22) we obtain

$$\begin{aligned}
\Delta_2 &= \max_{x_j, x_k \in Y} |v_k^* (\Phi(x_k - x_j) - \Phi_{\mathcal{P}_m}(x_k - x_j)) v_j| \\
&= \max_{x_j, x_k \in Y} \Big| \sum_{\rho,\sigma=1}^{s} \bar{v}_k(\rho) \Phi_{\rho,\sigma}(x_k - x_j) - \Phi_{\mathcal{P}_m \rho,\sigma}(x_k - x_j)) v_j(\sigma) \Big| \\
&\leq \max_{x_j, x_k \in Y} \Big\{ \|v_j\|_1 \|v_k\|_1 \frac{s^{k/2}}{k!} M^{\Phi}_{k,\nu} \|x_k - x_j\|_2^{k+\nu} \Big\},
\end{aligned}$$

and since x_j and x_k are both in W, we can estimate their distance by $2\sqrt{s}\delta$, which yields

$$\Delta_2 \leq \max_{x_j \in Y} \|v_j\|_1^2 \frac{s^{k/2}}{k!} M^{\Phi}_{k,\nu} (2\sqrt{s}\delta)^{k+\nu}. \tag{5.26}$$

Combining inequality (5.17), Theorem 5.5, and the upper bounds (5.23), (5.25), and (5.26), we obtain

$$\begin{aligned}
(P^{\eta}_{\Phi,\Lambda_Y})^2 &\leq \Delta_0 + 2\|c\|_1 \Delta_1 + \|c\|_1^2 \Delta_2 \\
&\leq 4 \left(\frac{l^2}{\delta}\right)^{|\alpha|} \|v\|_1^2 \max_{x_j \in Y} \|v_j\|_1 \frac{s^{(k-|\alpha|)/2}}{(k-|\alpha|)!} M^{\Phi}_{k,\nu} (2\sqrt{s}\delta)^{k+\nu-|\alpha|} \\
&\quad + 4 \left(\frac{l^2}{\delta}\right)^{2|\alpha|} \|v\|_1^2 \max_{x_j \in Y} \|v_j\|_1^2 \frac{s^{k/2}}{k!} M^{\Phi}_{k,\nu} (2\sqrt{s}\delta)^{k+\nu}.
\end{aligned}$$

Rearranging terms now leads to the estimate (5.19) of the power function. $\square$

Note that if we normalize $\|v\|_1 = 1$ and assume that $\max_{x_j \in X} \|v_j\|_1 = 1$, we obtain a simplified bound for the power function, which is similar to the bound for scalar-valued functions obtained in [23]. The result is stated in the following corollary.

Corollary 5.12. *If* $\|v\|_1 = 1$ *and* $\max_{x_j \in X} \|v_j\|_1 = 1$, *then Theorem 5.11 yields*

$$\begin{aligned}(P^{\eta}_{\Phi,\Lambda_Y})^2 \;&\leq\; 4M^{\Phi}_{k,\nu}\left(\frac{s^{(k-|\alpha|)/2}(2\sqrt{s}(m-1)^2)^{|\alpha|}}{(k-|\alpha|)!} + \frac{s^{k/2}(2\sqrt{s}(m-1)^2)^{2|\alpha|}}{k!}\right)\\ &\quad\times(2\sqrt{s}\delta)^{k+\nu-2|\alpha|},\end{aligned}$$

where $M^{\Phi}_{k,\nu}$ *is defined as in equation (5.24).*

This corollary does not show the dependency on the mesh norm explicitly. In Corollary 5.13, the results are given in terms of the mesh norm as desired.

2. Estimates on Ω

An immediate consequence of Theorem 5.11 is an error estimate on the whole Ω. We define the two expressions

$$\mathcal{C} := 2\left(\frac{(m-1)^2 h_{X,\Omega}}{\delta}\right)^{|\alpha|}, \tag{5.27}$$

$$\mathcal{R} := \frac{2\sqrt{s}\delta}{h_{X,\Omega}}. \tag{5.28}$$

Note that $\mathcal{C}$ is the upper bound on $\|c\|_1$ with the mesh norm scaled out if $\|v\|_1 = 1$. The second expression, $\mathcal{R}$, is the ratio of the diameter of the cube $W = W(w,\delta)$ and the mesh norm, $h_{X,\Omega}$. We now state an error estimate that holds on Ω.

Corollary 5.13. *Let* δ *satisfy* $\delta \geq 2(\sqrt{s}+s)(m-1)^2 h_{X,\Omega}$, *let* $\|v\|_1 = 1$, *and assume that* $\max_{x_j\in X}\|v_j\|_1 = 1$. *If* $W(w,\delta)$ *is contained in the domain* Ω, *and if* $x \in W(w,\delta)$, *then*

$$(P^{\eta}_{\Phi,\Lambda_Y})^2 \leq M^{\Phi}_{k,\nu}\left(2\mathcal{R}^{k+\nu-|\alpha|}\mathcal{C}\frac{s^{(k-|\alpha|)/2}}{(k-|\alpha|)!} + \mathcal{R}^{k+\nu}\mathcal{C}^2\frac{s^{k/2}}{k!}\right) h^{k+\nu-2\alpha}_{X,\Omega}, \tag{5.29}$$

where $\mathcal{R} = 4s(1+\sqrt{s})(m-1)^2$ *and* $\mathcal{C} = 2^{1-|\alpha|}(s+\sqrt{s})^{-|\alpha|}$, *and* $M^{\Phi}_{k,\nu}$ *is defined as in (5.24).*

Proof. Since $\delta \geq 2(\sqrt{s}+s)(m-1)^2 h_{X,\Omega}$ and $h_{Y,W} \leq (1+\sqrt{s})h_{X,\Omega}$ by inequality (5.14), we get $h_{Y,W} \leq \delta/[2\sqrt{s}(m-1)^2]$. Hence, the assumptions for Theorem 5.11 are fulfilled and (5.19) holds. Since $W \subset \Omega$, and since $Y \subset X$ implies that $\Lambda_Y \subset \Lambda$, we get $P^{\eta}_{\Phi,\Lambda} \leq P^{\eta}_{\Phi,\Lambda_Y}$, and we can replace Λ_Y by Λ and W by Ω in (5.19) to obtain inequality (5.29). $\square$

Remark 5.14. Corollary 5.13 says that for any x in the domain $\Omega \in \mathbb{R}^s$ that can be put in a cube $W(w,\delta)$ with side length $2\delta = 4(\sqrt{s}+s)(m-1)^2 h_{X,\Omega}$ which is completely contained in Ω, the power function can be bounded uniformly as given in inequality (5.19).

Remark 5.15. The expression $\mathcal{R}$ depends on the space dimension, s, and on the side length, δ, but not on the derivative order, $|\alpha|$. The ratio $\mathcal{C}$ depends on the degree of the polynomial and the side length of the cube, but not on the number of centers, N. Also, note that $\mathcal{C}$ decreases if δ increases. Since we can choose δ to be at least $2(\sqrt{s}+s)(m-1)^2 h_{X,\Omega}$, we can improve the estimate by choosing δ as large as possible but such that all cubes of interest are still contained in Ω, in case $|\alpha| > 1$.

Remark 5.16. The bounds obtained are consistent with classical results.

Finally, combining Corollary 5.13 with (5.4), we obtain a uniform error estimate on Ω.

Proposition 5.17. *If the assumptions of Proposition 5.1, Theorem 5.11, and Corollary 5.13 are fulfilled, then*

$$\sup_{x \in W(\omega,\delta) \subset \Omega} \|D^{\alpha}(f - s_f)\,(x)\|_{\infty} \leq |f|_{\Phi}\, \mathcal{T}^{1/2}\, h_{X,\Omega}^{(k+\nu-2\alpha)/2}, \tag{5.30}$$

where $\|g\|_\infty := \sup_{v\in\mathbb{R}^s,\|v\|_1=1}|v^* g\ (x)|$, *and*

$$\begin{aligned}
\mathcal{T} &:= M^{\Phi}_{k,\nu}\left(2\mathcal{R}^{k+\nu-|\alpha|}\mathcal{C}\frac{s^{(k-|\alpha|)/2}}{(k-|\alpha|)!}+\mathcal{R}^{k+\nu}\mathcal{C}^2\frac{s^{k/2}}{k!}\right),\\
\mathcal{R} &= 4s(1+\sqrt{s})(m-1)^2,\\
\mathcal{C} &= \frac{2^{1-|\alpha|}}{(s+\sqrt{s})^{|\alpha|}},
\end{aligned}$$

and the expression $M^{\Phi}_{k,\nu}$ *is defined as in (5.24).*

D. Examples

To conclude this chapter we investigate some examples of radial basis functions and show that they are in some class of Hölder continuous functions. Let us start with functions from $\mathbb{R}$ to $\mathbb{R}$.

Proposition 5.18. *Let* $\psi_{1,1}(x) := (1-|x|)^3_+(3|x|+1)$ *be the* C^2*-Wendland function for* $x \in \mathbb{R}$. *Then* $\psi_{1,1} \in C^1_1(\mathbb{R})$.

Proof. Assume $x \in [0,1]$. Then $\psi_{1,1}(x) = (1-x)^3(3x+1)$. But this implies that $\psi'_{1,1}(x) = -12x + 24x^2 - 12x^3$. Therefore, we get

$$\begin{aligned}
|\psi'_{1,1}(x)| &= |-12x^3+24x^2-12x| = 12|x||x^2-2x+1|\\
&= 12|x|(x-1)^2 \le 12|x|,
\end{aligned}$$

since $(x-1)^2 \le 1$ for $x \in [0,1]$. It is easy to verify that this is an upper bound for $\psi_{1,1}$ for all $x \in \mathbb{R}$. Hence, this yields that $\psi_{1,1} \in C^1_1(\mathbb{R})$. □

Proposition 5.19. *Let* $\psi_{1,2}(x) := (1-|x|)^5_+(8x^2+5|x|+1)$ *be the* C^4*-Wendland function for* $x \in \mathbb{R}$. *Then* $\psi_{1,2} \in C^3_1(\mathbb{R})$.

Proof. Assume $x \in [0,1]$. Then $\psi_{1,3}(x) = (1-x)^5(8x^2+5x+1)$. But this implies that $\psi_{1,3}^{(3)}(x) = -3360x^2 + 840x + 4200x^3 - 1680x^4$. Therefore, we get

$$\begin{aligned} |\psi_{1,3}^{(3)}(x)| &= |x||1680x^3 - 4200x^2 + 3360x - 840| \\ &= 840|x||2x-1|(x-1)^2 \leq 840|x|, \end{aligned}$$

since $|2x-1| \leq 1$ and $(x-1)^2 \leq 1$ for $x \in [0,1]$. It is easy to verify that this is an upper bound for $\psi_{1,3}$ for all $x \in \mathbb{R}$. Hence, this yields that $\psi_{1,3} \in C_1^3(\mathbb{R})$. $\square$

We now investigate functions from $\mathbb{R}^3$ to $\mathbb{R}$.

Proposition 5.20. *Let $\psi_{3,1}(x) := (1 - \|x\|_2)_+^4(4\|x\|_2 + 1)$ be the C^2-Wendland function for $x \in \mathbb{R}^3$. Then $\psi_{3,1} \in C_1^1(\mathbb{R}^3)$.*

Proof. Let us assume $\|x\| \leq 1$. We investigate the derivatives $D^\alpha\psi_{3,1}$, where $|\alpha| = 1$. Since $\partial_{x_i}\psi_{3,1} = 20(-1 + \|x\|^3)x_i$, for $1 \leq i \leq 3$, we get

$$|\partial_{x_i}\psi_{3,1}(x)| = 20|-1 + \|x\|^3||x_i| \leq 20\|x\|,$$

for $1 \leq i \leq 3$, since $|-1 + \|x\|^3| \leq 1$ and $|x_i| \leq \|x\|$ for all $1 \leq i \leq 3$. It is straightforward to verify that this is an upper bound for $\psi_{3,1}$ for all $x \in \mathbb{R}^3$. Hence, this yields that $\psi_{3,1} \in C_1^1(\mathbb{R}^3)$. $\square$

We conclude the chapter by looking at matrix-valued radial basis functions from $\mathbb{R}^2$ to $\mathbb{R}^{2\times 2}$.

Proposition 5.21. *Let $\psi_{3,2}(x) := (1-\|x\|_2)_+^6(35\|x\|_2^2+18\|x\|_2+3)/3$ be the Wendland function in C^4 for $x \in \mathbb{R}^2$. Let $\Phi := \{-\Delta I + \nabla\nabla^T\}\psi_{3,2}(x)$ be the 2×2 matrix-valued function based on the Wendland function $\psi_{3,2}$ with its components in C^2. Then $\Phi \in C_1^1(\mathbb{R}^2)$.*

Proof. Firstly, observe that for $\mathbf{x} = (x, y)^T$,

$$\Phi(\mathbf{x}) = \begin{pmatrix} -\partial_y^2 & \partial_x\partial_y \\ \partial_x\partial_y & -\partial_x^2 \end{pmatrix} \psi_{3,2}(\mathbf{x}) := \begin{pmatrix} \Phi_{1,1}(\mathbf{x}) & \Phi_{1,2}(\mathbf{x}) \\ \Phi_{2,1}(\mathbf{x}) & \Phi_{2,2}(\mathbf{x}) \end{pmatrix},$$

with

$$\begin{aligned} \Phi_{1,1}(\mathbf{x}) &= -56/3\ (\|\mathbf{x}\| - 1)_+^4 (35y^2 + 5x^2 - 1 - 4\|\mathbf{x}\|), \\ \Phi_{1,2}(\mathbf{x}) &= \Phi_{2,1}(\mathbf{x}) = 560\ (\|\mathbf{x}\| - 1)_+^4 xy, \text{ and} \\ \Phi_{2,2}(\mathbf{x}) &= -56/3\ (\|\mathbf{x}\| - 1)_+^4 (35x^2 + 5y^2 - 1 - 4\|\mathbf{x}\|). \end{aligned}$$

For $|\alpha| = 1$, we have to check the two derivatives $D^{(1,0)} = \partial_x$ and $D^{(0,1)} = \partial_y$. We get for $\|x\| \leq 1$:

$$\begin{aligned} \partial_x\Phi_{1,1}(\mathbf{x}) &= -560\ (\|\mathbf{x}\| - 1)^3 x(5y^2 + x^2 - \|\mathbf{x}\|)/\|\mathbf{x}\| \\ \partial_x\Phi_{1,2}(\mathbf{x}) &= 560\ (\|\mathbf{x}\| - 1)^3 y(5x^2 + y^2 - \|\mathbf{x}\|)/\|\mathbf{x}\| \\ \partial_x\Phi_{2,2}(\mathbf{x}) &= -560\ (\|\mathbf{x}\| - 1)^3 x(7x^2 + 3y^2 - 3\|\mathbf{x}\|)/\|\mathbf{x}\|, \end{aligned}$$

and

$$\begin{aligned} \partial_y\Phi_{1,1}(\mathbf{x}) &= -560\ (\|\mathbf{x}\| - 1)^3 y(7y^2 + 3x^2 - 3\|\mathbf{x}\|)/\|\mathbf{x}\| \\ \partial_y\Phi_{1,2}(\mathbf{x}) &= 560\ (\|\mathbf{x}\| - 1)^3 x(5y^2 + x^2 - \|\mathbf{x}\|)/\|\mathbf{x}\| \\ \partial_y\Phi_{2,2}(\mathbf{x}) &= -560\ (\|\mathbf{x}\| - 1)^3 y(5x^2 + y^2 - \|\mathbf{x}\|)/\|\mathbf{x}\|. \end{aligned}$$

Let us investigate $\partial_x\Phi_{1,1}(\mathbf{x})$. Our goal is to find an upper bound for the derivative. We have

$$\begin{aligned} |\partial_x\Phi_{1,1}(\mathbf{x})| &\leq 560\ |\|\mathbf{x}\| - 1|^3\ |x|\ \left|5y^2 + x^2 - \|\mathbf{x}\|\right| / \|\mathbf{x}\| \\ &\leq 560\ \left|5y^2 + x^2 - \|\mathbf{x}\|\right|, \end{aligned}$$

since $|\|\mathbf{x}\| - 1|^3 \leq 1$ and $|x| \leq \|\mathbf{x}\|$. We now obtain

$$\begin{aligned} |\partial_x \Phi_{1,1}(\mathbf{x})| &\leq 560\ \left(5y^2 + x^2 + \|\mathbf{x}\|\right) \leq 560\ \left(5y^2 + 5x^2 + \|\mathbf{x}\|\right) \\ &= 560\ \left(5\|\mathbf{x}\|^2 + \|\mathbf{x}\|\right) \leq 3360\ \|\mathbf{x}\|, \end{aligned}$$

since $\|\mathbf{x}\|^2 \leq \|\mathbf{x}\|$ for $\|\mathbf{x}\| \leq 1$. Similarly, we can obtain upper bounds for the other derivatives. Therefore, since it is easy to verify that this gives upper bounds for all components of $\Phi(\mathbf{x})$ for all $x \in \mathbb{R}$, we obtain that $\Phi \in C_1^1(\mathbb{R}^2)$. $\square$

CHAPTER VI

STABILITY ESTIMATES

In this chapter we investigate the stability of the interpolation matrix based on the matrix-valued RBF which is done via a study of condition numbers. Stability determines how much the interpolant changes in case of (small) perturbations of the data. We derive estimates on the norm of the inverse of the interpolation matrix which arises in a broad class of multivariate Hermite interpolation problems. It turns out that these estimates depend on the same parameters as in the case of ordinary interpolation with scalar-valued radial basis functions [20, 21]. More explicitly, the stability estimates depend on the particular generating scalar-valued radial basis function, the space dimension, and the minimal separation of the Hermite data. It does not depend on the number or distribution of the data sites. We expect similar methods as those derived in this chapter to lead to stability estimates for a broader class of Hermite interpolation problems. The approach we take here is based on [22, Section 7].

A. Multivariate Hermite Interpolation

We investigate upper bounds for the norm of the inverse of the interpolation matrix A which is based on the following class of multivariate Hermite interpolation problems: Given a set of distinct points $X := \{x_j\}_{j=1}^N$ in $\mathbb{R}^s$ with minimum separation defined by $2q := \min_{j \neq k} \|x_j - x_k\|$, we want to interpolate a vector-valued function defined on $\mathbb{R}^s$ at given function values and certain derivative information of first order obtained at the data sites X. More specifically, the distributions generating the data are given to be δ-functionals and first order derivatives of δ-functionals. Hence, we set the linear functionals $\Lambda := \{\lambda_j\}_{j=1}^N \cup \{\eta_j\}_{j=1}^N$, to be given as

$$\lambda_j := v_j \, \delta_{x_j} \text{ and } \eta_j := w_j \, D^{\alpha_j}\delta_{x_j}, \tag{6.1}$$

where $|\alpha_j| = 1$, for $1 \leq j \leq N$. Here, $v_j, w_j \in \mathbb{R}^s$ are arbitrary vectors of fixed length $\|v_j\| = \|w_j\| = L \geq 1/12$ for all j's. This leads to more general results which can be standardized to have unit length by setting $L = 1$. Let the matrix-valued basis functions have the form

$$\Phi(x) = \int\limits_{\mathbb{R}^s} e^{ix\cdot\xi} \, d\mu(\xi), \tag{6.2}$$

with $d\mu(\xi) := (2\pi)^{-s}\|\xi\|^2\Pi(\xi)\hat{\psi}(\xi)d\xi$, where $\Pi(\xi)$ is defined to be the projection $\Pi(\xi) := I - \|\xi\|^{-2}\xi\xi^T$ for $\xi \neq 0$ as defined in Chapter III and let $\hat{\psi}(\xi)$ be the Fourier transform of a scalar-valued radial basis function $\psi \in$ SPD. Hence, Φ is an order-0 $\mathcal{S}$-CPD function.

We assume that Φ is strictly $\mathcal{S}$-CPD and that the generating scalar-valued RBF ψ has a (general) Fourier transform which is positive, i.e.

$$\hat{\psi}(\xi) > 0 \text{ for all } \xi \in \mathbb{R}^s. \tag{6.3}$$

Then the measure $d\mu(\xi)$ is a positive measure whose support contains an open subset of $\mathbb{R}^s$. The scalar-valued radial basis functions of interest fulfill this assumption [25, 29], as for example the Gaussian functions, $\psi_\alpha(x) = e^{-\alpha\|x\|^2}$, with $\alpha > 0$, and the class of Wendland functions $\psi_{l,k}(x)$ defined in Chapter III, for $l \geq 2$.

B. Estimates for the Quadratic Form

We want to obtain upper bounds for the inverse of the $2N \times 2N$ interpolation matrix A arising from the interpolation problem described above. Our goal is to show that

$\|A^{-1}\| \leq \theta(q,s)$ for some function θ independent of the number of data sites, N. Note that θ depends only on the separation distance, q, the space dimension, s, and on the RBF. Upper bounds for matrix-valued functions generated by other RBFs can be obtained using ideas employed in [21].

Note that the matrix A is positive definite when Φ is $\mathcal{S}$-CPD. Hence, estimating $\|A^{-1}\|$ amounts to obtaining a lower bound on the lowest eigenvalue of A and then taking inverses. We now investigate the quadratic form c^*Ac, where $c = (a_1, \ldots, a_N, b_1, \ldots, b_N)^T \in \mathbb{R}^{2N}$, in order to obtain an estimate for the lowest eigenvalue. Here, (2.10) and (6.2) yield

$$\begin{aligned} c^*Ac &= (l^* \otimes l, \Phi) = \int_{\mathbb{R}^s} l(x)^* (\Phi * l)(x)\, dx \\ &= \int_{\mathbb{R}^s} l(x)^* \int_{\mathbb{R}^s}\int_{\mathbb{R}^s} e^{i(x-y)\cdot\xi} d\mu(\xi)\, l(y)\, dy\, dx \\ &= \int_{\mathbb{R}^s} \left(\int_{\mathbb{R}^s} e^{-ix\cdot\xi} l(x)\, dx \right)^* d\mu(\xi) \left(\int_{\mathbb{R}^s} e^{-iy\cdot\xi} l(y)\, dy \right) \\ &= \int_{\mathbb{R}^s} \hat{l}(\xi)^* d\mu(\xi)\, \hat{l}(\xi) =: \|\hat{l}\|_\mu^2, \end{aligned} \tag{6.4}$$

where $l(x) := \sum_{j=1}^N a_j \lambda_j(x) + \sum_{j=1}^N b_j \eta_j(x)$. If $d\nu(\xi)$ is any positive measure such that its support contains an open subset of $\mathbb{R}^s$ fulfilling $d\mu(\xi) \geq d\nu(\xi) \geq 0$, then

$$c^*Ac = \|\hat{l}\|_\mu^2 \geq \|\hat{l}\|_\nu^2 =: c^*Bc,$$

and hence,

$$c^*Ac \geq \sum_{j,k=1}^N (\bar{a}_j\ a_k \langle \hat{\lambda}_k, \hat{\lambda}_j\rangle_\nu + \bar{b}_j\ b_k \langle \hat{\eta}_k, \hat{\eta}_j\rangle_\nu + \bar{a}_j\ b_k \langle \hat{\eta}_k, \hat{\lambda}_j\rangle_\nu + \bar{b}_j\ a_k \langle \hat{\lambda}_k, \hat{\eta}_j\rangle_\nu).$$

If we split the sum in the parts where $j = k$ and $j \neq k$ and use $|a|^2 + |b|^2 \geq 2|ab|$, we obtain the bound

$$
\begin{aligned}
c^*Ac \quad \geq \quad & \sum_{j=1}^{N} |a_j|^2 \left(\|\hat{\lambda}_j\|_\nu^2 - \sum_{k \neq j} |\langle \hat{\lambda}_k, \hat{\lambda}_j \rangle_\nu| - \sum_{k=1}^{N} |\langle \hat{\lambda}_k, \hat{\eta}_j \rangle_\nu| \right) \\
& + \sum_{j=1}^{N} |b_j|^2 \left(\|\hat{\eta}_j\|_\nu^2 - \sum_{k \neq j} |\langle \hat{\eta}_k, \hat{\eta}_j \rangle_\nu| - \sum_{k=1}^{N} |\langle \hat{\lambda}_k, \hat{\eta}_j \rangle_\nu| \right).
\end{aligned}
\tag{6.5}
$$

We now investigate the measure $d\nu(\xi)$. Let $d\nu(\xi) := \|\xi\|^2 \Pi(\xi) \hat{\chi}\xi \; d\xi$, where, since $d\mu(\xi) = \|\xi\|^2 \Pi(\xi) \hat{\psi}(\xi) \; d\xi$, we want that $\hat{\psi}(\xi) \geq \hat{\chi}(\xi) \geq 0$ and that $\hat{\chi}$ is compactly supported with its support containing some open subset of $\mathbb{R}^s$. The inverse Fourier transform of $\hat{\chi}$ is $(2\pi)^s \chi$. Since $\hat{\chi}$ has compact support, its inverse Fourier transform χ is an entire function, and hence any differentiation of χ is well-defined. Our next goal is to compute the forms $\langle \cdot, \cdot \rangle_\nu$ occurring in inequality (6.5). We can write the general form of the considered linear functionals as $\tilde{\lambda}_j := \tilde{v}_j \; D^{\tilde{\alpha}_j} \delta_{x_j}$, with $|\tilde{\alpha}_j| \leq 1$ and $\|\tilde{v}_j\| = L$ for $1 \leq j \leq 2N$, such that $\tilde{\lambda}_j = \lambda_j$ for $1 \leq j \leq N$ and $\tilde{\lambda}_j = \eta_j$ for $N+1 \leq j \leq 2N$. Hence,

$$
\begin{aligned}
\langle \hat{\tilde{\lambda}}_j^*, \hat{\tilde{\lambda}}_k \rangle_\nu &= \int_{\mathbb{R}^s} (\tilde{v}_j \widehat{D^{\tilde{\alpha}_j}(\xi)} \delta_{x_j})^* \, d\nu(\xi) \, (\tilde{v}_k \widehat{D^{\tilde{\alpha}_k} \delta_{x_k}})(\xi) \\
&= \int_{\mathbb{R}^s} \left(\int_{\mathbb{R}^s} e^{-ix\cdot\xi} \tilde{v}_j D^{\tilde{\alpha}_j} \delta_{x_j}(x) \, dx \right)^* d\nu(\xi) \left(\int_{\mathbb{R}^s} e^{-iy\cdot\xi} \tilde{v}_k D^{\tilde{\alpha}_k} \delta_{x_k}(y) \, dy \right) \\
&= \int_{\mathbb{R}^s} \tilde{v}_j^* D^{\tilde{\alpha}_j} \delta_{x_j}(x) \, dx \int_{\mathbb{R}^s} e^{i(x-y)\cdot\xi} \, d\nu(\xi) \int_{\mathbb{R}^s} \tilde{v}_k D^{\tilde{\alpha}_k} \delta_{x_k}(y) \, dy \\
&= \int_{\mathbb{R}^s} \int_{\mathbb{R}^s} \int_{\mathbb{R}^s} \tilde{v}_j^* D^{\tilde{\alpha}_j} \delta_{x_j}(x) e^{i(x-y)\cdot\xi} \{\|\xi\|^2 I - \xi\xi^T\} \hat{\chi}(\xi) \tilde{v}_k D^{\tilde{\alpha}_k} \delta_{x_k}(y) \, dx \, dy \, d\xi \\
&= \int_{\mathbb{R}^s} (i\xi)^{\tilde{\alpha}_j + \tilde{\alpha}_k} (-1)^{\tilde{\alpha}_k} e^{i(x_j - x_k)\cdot\xi} \, \tilde{v}_j^* \{\|\xi\|^2 I - \xi\xi^T\} \tilde{v}_k \, \hat{\chi}(\xi) \, d\xi \\
&= (-1)^{\tilde{\alpha}_k} (2\pi)^s \, \tilde{v}_j^* [(D^{\tilde{\alpha}_j + \tilde{\alpha}_k} \{-\Delta I + \nabla\nabla^T\} \chi)(x_j - x_k)] \tilde{v}_k,
\end{aligned}
$$

where we used Fourier transform arguments to calculate the derivatives. Applying

$\|\tilde{v}_j\| = L$ and using that $\|A\|_2 = \sup_{x,y\neq 0}|y^*Ax|/(\|x\|\|y\|)$ for a matrix A gives

$$|\langle\hat{\tilde{\lambda}}_j^*, \hat{\tilde{\lambda}}_k\rangle_\nu| \leq (2\pi)^s L^2 \|(D^{\tilde{\alpha}_j+\tilde{\alpha}_k}\{-\Delta I + \nabla\nabla^T\}\chi)(x_j - x_k)\|_2, \tag{6.6}$$

where $\|\cdot\|_2$ is the associated matrix norm in l^2, i.e. the spectral norm. This leads to the following lower bound for the quadratic form

$$\begin{aligned} c^*Ac \;&\geq\; (2\pi)^s \sum_{j=1}^{2N} |a_j|^2 \Bigg(|\tilde{v}_j^*\ (D^{2\tilde{\alpha}_j}\{-\Delta I + \nabla\nabla^T\}\chi)(0)\ \tilde{v}_j| \\ &\qquad - \sum_{k\neq j} |\tilde{v}_j^*\ (D^{\tilde{\alpha}_j+\tilde{a}_k}\{-\Delta I + \nabla\nabla^T\}\chi)(x_j - x_k)\ v_k| \Bigg) \\ &\geq\; (2\pi)^s \sum_{j=1}^{2N} |a_j|^2 \Bigg(|\tilde{v}_j^*\ (D^{2\tilde{\alpha}_j}\{-\Delta I + \nabla\nabla^T\}\chi)(0)\ \tilde{v}_j| \\ &\qquad - L^2 \sum_{k\neq j} \|D^{\tilde{\alpha}_j+\tilde{a}_k}\{-\Delta I + \nabla\nabla^T\}\chi)(x_j - x_k)\|_2 \Bigg), \end{aligned} \tag{6.7}$$

where we applied inequality (6.6) in the last step. Note that $2|\tilde{\alpha}_j| \in \{0, 2\}$ and that $0 \leq |\tilde{\alpha}_j + \tilde{\alpha}_k| \leq 2$. That means that we have to determine derivatives of χ up to fourth order.

1. Choice of the Function χ

We now choose the function χ properly such that we get good results and

$$\hat{\psi}(\xi) \geq \hat{\chi}(\xi) \geq 0 \tag{6.8}$$

holds. For an even integer p define the function

$$\mathcal{B}_p := \underbrace{\chi_{1/2} * \ldots * \chi_{1/2}}_{p\text{-fold}},$$

where $\chi_{1/2}$ is the characteristic function with support $[-1/2, 1/2]$. Note that $\mathcal{B}_p$ is a function from $\mathbb{R}$ to $\mathbb{R}$. Define the tensor product spline as $T_p(x) := \prod_{t=1}^{s} \mathcal{B}_p(x(t))$, where $x(t)$ is the t^{th} component of $x \in \mathbb{R}^s$. Then [3] gives:

(a) T_p is an even, piecewise polynomial function;

(b) $T_p(x) > 0$ for every x in $C = \{x : -p/2 < x(t) < p/2, 1 \leq t \leq s\}$, zero elsewhere;

(c) $T_p(0) = \mathcal{B}_p(0)^s \geq T_p(x)$ for all $x \in \mathbb{R}^s$;

(d) $\hat{T}_p(\xi) = \prod_{t=1}^{s} \operatorname{sinc}^p(\xi(t))$;

(e) $T_p = \left\{(2\pi)^{-s} \prod_{t=1}^{s} \operatorname{sinc}^p(x(t))\right\}^{\wedge}$.

Here, the equation (e) is obtained from (d) and Fourier transform properties. Combining these properties of T_p yields the following lemma. Property (b) means that the support of T_p is the s-dimensional cube with sides $[-p/2, p/2]$.

Lemma 6.1. *Let γ be any positive number and p any even integer, define*

$$\varphi_0(r) := \inf_{\|\xi\| \leq 2r} \hat{\psi}(\xi) \geq 0 \text{ and} \tag{6.9}$$

$$c_\gamma := \frac{\varphi_0(\frac{p}{\gamma})}{\mathcal{B}_p(0)^s (2\pi\gamma)^s}. \tag{6.10}$$

Then $\hat{\psi}(\xi) \geq \hat{\chi_\gamma}(\xi) \geq 0$ for any member of the family of functions defined by

$$\hat{\chi}_\gamma(\xi) := (2\pi\gamma)^s c_\gamma T_p(\gamma\xi), \tag{6.11}$$

and the inverse Fourier transform of (6.11) is given by

$$\chi_\gamma(x) = c_\gamma \prod_{t=1}^{s} \operatorname{sinc}^p(x(t)/\gamma), \tag{6.12}$$

where $x(k)$ is the k^{th} component of $x \in \mathbb{R}^s$.

2. Certain Derivatives of χ_γ

We now want to estimate the terms arising in inequality (6.7), where χ_γ is given as in equation (6.12). We therefore need estimates for the entries of the $s \times s$ dimensional matrix $P(x) := D^{\tilde{\alpha}_j+\tilde{\alpha}_k}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x)$. In order to obtain these bounds, we begin with a calculation of $D^\alpha \chi_\gamma(x)$, where $1 \le |\alpha| \le 2$. If $|\alpha| = 1$, i.e. for the first partial derivatives, we obtain

$$\begin{aligned} \partial_{x(j)}\chi_\gamma(x) &= c_\gamma \partial_{x(j)}\left(\prod_{t=1}^{s} \mathrm{sinc}^p(x(t)/\gamma)\right) \\ &= \frac{c_\gamma p}{\gamma}\prod_{t\neq j} \mathrm{sinc}^p(x(t)/\gamma)\mathrm{sinc}^{p-1}(x(j)/\gamma)\mathrm{sinc}'(x(j)/\gamma). \end{aligned} \tag{6.13}$$

If $j \neq k$, differentiating (6.13) gives the following partials of second order

$$\begin{aligned} \partial_{x(k)}\partial_{x(j)}\chi_\gamma(x) &= \frac{c_\gamma p^2}{\gamma^2}\prod_{t\neq j,k} \mathrm{sinc}^p(x(t)/\gamma) \\ &\quad\cdot \prod_{r=j,k} \mathrm{sinc}^{p-1}(x(r)/\gamma)\mathrm{sinc}'(x(r)/\gamma). \end{aligned} \tag{6.14}$$

If $j = k$, we obtain the following partials:

$$\begin{aligned} \partial^2_{x(j)}\chi_\gamma(x) &= \frac{c_\gamma p}{\gamma^2}\prod_{t\neq j} \mathrm{sinc}^p(x(t)/\gamma)\Big\{(p-1)\mathrm{sinc}^{p-2}(x(j)/\gamma)\mathrm{sinc}'(x(j)/\gamma)^2 \\ &\quad + \mathrm{sinc}^{p-1}(x(j)/\gamma)\mathrm{sinc}''(x(j)/\gamma)\Big\}. \end{aligned} \tag{6.15}$$

If we define the matrix P to have entries $P_{jk}(x) := \{-\delta_{jk}\sum_{l=1}^{s}\partial^2_{x(l)} + \partial_{x(j)}\partial_{x(k)}\}\chi_\gamma(x)$, for $1 \le j,k \le s$, combining the results obtained in (6.14) and (6.15) yields for $j = k$:

$$\begin{aligned} P_{jj}(x) &= -\frac{c_\gamma p}{\gamma^2}\sum_{l\neq j}\prod_{t\neq l} \mathrm{sinc}^p(x(t)/\gamma) \\ &\quad\cdot\Big\{(p-1)\mathrm{sinc}^{p-2}(x(l)/\gamma)\mathrm{sinc}'(x(l)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(l)/\gamma)\mathrm{sinc}''(x(l)/\gamma)\Big\} \end{aligned} \tag{6.16}$$

and, for the off-diagonal entries, i.e. for $j \neq k$, we get

$$P_{jk}(x) = \frac{c_\gamma p^2}{\gamma^2} \prod_{t \neq j,k} \mathrm{sinc}^p(x(t)/\gamma) \prod_{r=j,k} \mathrm{sinc}^{p-1}(x(r)/\gamma)\mathrm{sinc}'(x(r)/\gamma). \tag{6.17}$$

We now state the following result for the elements $P_{jk}(0)$, for $1 \leq j, k \leq s$.

Lemma 6.2. *If $x = 0$ and $|\alpha| = 0$, then*

$$\Big(\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(0)\Big)_{jk} = \begin{cases} 0 & \text{for } j \neq k, \\ \dfrac{c_\gamma p\,(s-1)}{3\gamma^2} & \text{for } j = k. \end{cases} \tag{6.18}$$

Proof. Using the fact that $\mathrm{sinc}(0) = 1$, $\mathrm{sinc}'(0) = 0$, $\mathrm{sinc}''(0) = -1/3$ and applying equations (6.16) and (6.17) for $x = 0$ yields the result. □

In order to be able to estimate derivatives up to fourth order of the function $\mathrm{sinc}(x)$, we now state a result that was obtained in [22, Lemma 7.2].

Lemma 6.3. *There holds*

$$|\mathrm{sinc}^{(l)}(x)| \leq \min\{1, \frac{2}{|x|}\} \text{ for all } l \geq 0 \text{ and } x \neq 0. \tag{6.19}$$

It remains to obtain the values for the entries $\Big((D^\alpha\{-\Delta I + \nabla\nabla^T\}\chi_\gamma)(0)\Big)_{jk}$, where $1 \leq |\alpha| \leq 2$, and if $x \neq 0$ to get estimates for the entries of the matrix $D^\alpha\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x)$, where $0 \leq |\alpha| \leq 2$. The proof of the following lemma is given in the appendix.

Lemma 6.4. *For $0 \leq |\alpha| \leq 2$ and $x \neq 0$, there holds*

$$\|D^\alpha\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x)\|_\infty \leq 2c_\gamma(s-1)\left(\frac{p}{\gamma}\right)^{|\alpha|+2} \prod_{t=1}^{s}\left[\min\left\{1, \frac{2\gamma}{|x(t)|}\right\}\right]^p.$$

Suppose that $x \neq 0$ in the inequality of Lemma 6.4. Let $|x(t_0)| = \max_t |x(t)|$. Standard arguments give $\|x\|_\infty = |x(t_0)| \geq 1/\sqrt{s}\,\|x\|_2$. Use $\min\{1, 2\gamma/|x(t)|\} \leq 1$ if

$t \neq t_0$ and $\min\{1, 2\gamma/|x(t_0)|\} \leq 2\gamma/|x(t_0)| \leq 2\gamma\sqrt{s}/\|x\|_2$ if $t = t_0$, as in [22]. Note also that $\|A\|_2 \leq \sqrt{s}\|A\|_\infty$ for a matrix A. Using the inequality of Lemma 6.4 for $0 \leq |\alpha| \leq 2$ and $x \neq 0$ yields the estimate

$$\begin{aligned} \|D^\alpha\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x)\|_2 &\leq \sqrt{s}\,\|D^\alpha\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x)\|_\infty \\ &\leq 2c_\gamma\sqrt{s}(s-1)\left(\frac{p}{\gamma}\right)^{|\alpha|+2}\left[\frac{2\gamma\sqrt{s}}{\|x\|_2}\right]^p. \end{aligned} \tag{6.20}$$

We use this inequality in order to obtain the lower bounds for the quadratic form (6.7).

As a consequence of the proof of Lemma 6.4 in Appendix A, we are now able to evaluate $\Big((D^\alpha\{-\Delta I + \nabla\nabla^T\}\chi_\gamma)(0)\Big)_{jk}$ for $1 \leq |\alpha| \leq 2$. Using the fact that $\mathrm{sinc}(0) = 1$, $\mathrm{sinc}'(0) = 0$, $\mathrm{sinc}''(0) = -1/3$, and $\mathrm{sinc}'''(0) = 0$, we obtain the following result:

Lemma 6.5. *If $|\alpha| = 1$ and $x = 0$, then*

$$\Big((D^\alpha\{-\Delta + \nabla\nabla^T\}\chi_\gamma)(0)\Big)_{jk} = 0 \text{ for all } 1 \leq j, k \leq N. \tag{6.21}$$

Let $\alpha = \alpha_l + \alpha_i$. Then, if $|\alpha| = 2$ and $x = 0$, there holds

$$\begin{aligned} &\Big((D^{\alpha_l+\alpha_i}\{-\Delta + \nabla\nabla^T\}\chi_\gamma)(0)\Big)_{jk} \\ &= \begin{cases} -c_\gamma \frac{p}{9\gamma^4}\Big\{p(s-2) + 3(p-1)\Big\} & \text{for } l = i \neq j, \text{ and } j = k, \\ -c_\gamma \frac{p^2}{9\gamma}(s-1) & \text{for } l = i = j = k, \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{6.22}$$

C. Stability Theorem and Proof

Combining the results of the earlier sections of this chapter allows us to prove the following theorem yielding a lower bound for the quadratic form c^*Ac:

Theorem 6.6. *For space dimension $s \geq 2$, let $p \geq s+1$ be an even integer, and choose $\gamma = \gamma(q,p,s)$ such that $\gamma > 0$ and it satisfies the inequalities*

$$3\gamma \leq 1 \quad \text{and} \quad 3^{s+2}p\sqrt{s}(2\gamma\sqrt{s}q^{-1})^p \left[1 + \frac{p}{\gamma}\right] \leq \frac{1}{4}. \tag{6.23}$$

If $\Phi(x)$ has the form

$$\Phi(x) = \int_{\mathbb{R}^s} e^{ix\cdot\xi}\, d\mu(\xi),$$

with $d\mu(\xi) := (2\pi)^{-s}\|\xi\|^2\Pi(\xi)\hat{\psi}(\xi)\, d\xi$, where $\Pi(\xi) := I - \|\xi\|^{-2}\xi\xi^T$ for $\xi \neq 0$ and $\hat{\psi}$ is the positive Fourier transform of a scalar-valued RBF ψ such that Φ is an order-0 strictly $\mathcal{S}$-CPD function, then

$$c^*Ac \geq \theta(q,s,p)\|c\|^2, \text{ where } \theta := \frac{1}{2}\frac{\varphi_0(\frac{p}{\gamma})}{\mathcal{B}_p(0)^s\gamma^s}, \tag{6.24}$$

and φ_0 is defined as in (6.9).

Proof. Combining (6.5) and (6.6) gives

$$\begin{aligned} c^*Ac \;\geq\; & \sum_{j=1}^{N}|a_j|^2(2\pi)^s\Big(|v_j^*\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(0)v_j| \\ & \quad - L^2\sum_{k\neq j}\|\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x_j - x_k)\|_2 \\ & \quad - L^2\sum_{k=1}^{N}\|D^{\alpha_k}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x_j - x_k)\|_2\Big) \\ & + \sum_{j=1}^{N}|b_j|^2(2\pi)^s\Big(|w_j^*\; D^{2\alpha_j}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(0)w_j| \\ & \quad - L^2\sum_{k\neq j}\|D^{\alpha_j+\alpha_k}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x_j - x_k)\|_2 \\ & \quad - L^2\sum_{k=1}^{N}\|D^{\alpha_j}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(x_j - x_k)\|_2\Big). \end{aligned} \tag{6.25}$$

We first investigate the term $|w_j^*\; D^{2\alpha_j}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(0)w_j|$. Denote the matrix

$D^{2\alpha_j}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(0) =: M(0)$. If we now apply (6.22) we obtain

$$\begin{aligned}
|w_j^* \, D^{2\alpha_j}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(0) w_j| &= \sum_{i=1}^{N} |w_j(i)|^2 |M_{ii}(0)| \\
&\geq \min_j |M_{jj}(0)| \sum_{i=1}^{N} |w_j(i)|^2 \\
&= c_\gamma \frac{p^2}{9\gamma^4}(s-1)L^2,
\end{aligned}$$

since the diagonal entries are the only non-zero entries of $D^{2\alpha_j}\{-\Delta I + \nabla\nabla^T\}\chi_\gamma(0)$ and since $p(s-2) + 3(p-1) \geq p(s-1) \geq 0$. Applying (6.18) and (6.20) now yields the estimate

$$\begin{aligned}
\frac{c^* A c}{(2\pi)^s c_\gamma} \geq \; & \sum_{j=1}^{N} |a_j|^2 \Bigg\{ L^2 \frac{p}{3\gamma^2}(s-1) - 2L^2\sqrt{s}(2\gamma\sqrt{s})^p (s-1) \left(\frac{p}{\gamma}\right)^2 \sum_{k \neq j} \frac{1}{\|x_j - x_k\|_2^p} \\
& \qquad - 2L^2\sqrt{s}(2\gamma\sqrt{s})^p (s-1) \left(\frac{p}{\gamma}\right)^3 \sum_{k \neq j} \frac{1}{\|x_j - x_k\|_2^p} \Bigg\} \\
& + \sum_{j=1}^{N} |b_j|^2 \Bigg\{ L^2 \frac{p^2}{9\gamma^4}(s-1) - 2L^2\sqrt{s}(2\gamma\sqrt{s})^p (s-1) \left(\frac{p}{\gamma}\right)^4 \sum_{k \neq j} \frac{1}{\|x_j - x_k\|_2^p} \\
& \qquad - 2L^2\sqrt{s}(2\gamma\sqrt{s})^p (s-1) \left(\frac{p}{\gamma}\right)^3 \sum_{k \neq j} \frac{1}{\|x_j - x_k\|_2^p} \Bigg\}.
\end{aligned}$$

Note that we replaced $\sum\limits_{k=1}^{N}$ by $\sum\limits_{k \neq j}$ because of (6.21). Grouping similar terms together gives the lower bound

$$\begin{aligned}\frac{c^*Ac}{(2\pi)^s c_\gamma} \quad \geq \quad & \sum_{j=1}^{N} |a_j|^2 \Bigg\{ L^2 \frac{p}{3\gamma^2}(s-1) - 2L^2\sqrt{s}(2\gamma\sqrt{s})^p(s-1)\left(\frac{p}{\gamma}\right)^2 \\ & \qquad \cdot \left[1+\frac{p}{\gamma}\right] \sum_{k\neq j} \frac{1}{\|x_j-x_k\|_2^p} \Bigg\} \\ & + \sum_{j=1}^{N} |b_j|^2 \Bigg\{ L^2 \frac{p^2}{9\gamma^4}(s-1) - 2L^2\sqrt{s}(2\gamma\sqrt{s})^p(s-1)\left(\frac{p}{\gamma}\right)^3 \\ & \qquad \cdot \left[1+\frac{p}{\gamma}\right] \sum_{k\neq j} \frac{1}{\|x_j-x_k\|_2^p} \Bigg\}. \end{aligned} \tag{6.26}$$

We now are left to estimate $\sum_{k\neq j} 1/\|x_j - x_k\|_2^p$. Here, we use the technique presented by Narcowich and Ward in [20] to get

$$\sum_{k\neq j} \frac{1}{\|x_j-x_k\|_2^p} \leq 3^s \sum_{n=1}^{\infty} n^{s-1}\kappa_n,$$

where

$$\kappa_n := \sup\{\|x\|^{-p} : nq \leq \|x\| \leq (n+1)q\}$$

and $2q$ is the minimal separation distance of the data sites $X = \{x_j\}_{j=1}^N$. Clearly, we have that $\kappa_n = (nq)^{-p}$ holds. Therefore,

$$\sum_{k\neq j} \frac{1}{\|x_j-x_k\|_2^p} \leq 3^s q^{-p} \sum_{n=1}^{\infty} n^{s-p-1} \leq 3^s q^{-p} \sum_{n=1}^{\infty} n^{-2},$$

since $p \geq s+1$. Using the fact that the last sum is equal to $\pi^2/6$, we hence obtain

$$\sum_{k\neq j} \frac{1}{\|x_j-x_k\|_2^p} \leq 3^s q^{-p}\frac{\pi^2}{6} \leq 3^{s+1} q^{-p}. \tag{6.27}$$

Combining (6.26) and (6.27) now yields the lower bound for c^*Ac of the form

$$\begin{aligned}\frac{c^*Ac}{(2\pi)^s c_\gamma} \quad &\geq \quad \sum_{j=1}^{N} |a_j|^2 (s-1) L^2 \frac{p}{3\gamma^2} \left\{ 1 - \frac{6\gamma^2}{p} \sqrt{s} (2\gamma\sqrt{s} q^{-1})^p \, 3^{s+1} \left(\frac{p}{\gamma}\right)^2 \left[1 + \frac{p}{\gamma}\right] \right\} \\ &\quad + \sum_{j=1}^{N} |b_j|^2 (s-1) L^2 \frac{p^2}{9\gamma^4} \left\{ 1 - 3\gamma \frac{6\gamma^2}{p} \sqrt{s} (2\gamma\sqrt{s} q^{-1})^p \, 3^{s+1} \left(\frac{p}{\gamma}\right)^2 \left[1 + \frac{p}{\gamma}\right] \right\}.\end{aligned}$$

Clearly, we are able to choose $\gamma > 0$ such that

$$3\gamma \leq 1 \quad \text{and} \quad \frac{6\gamma^2}{p} \sqrt{s} (2\gamma\sqrt{s} q^{-1})^p \, 3^{s+1} \left(\frac{p}{\gamma}\right)^2 \left[1 + \frac{p}{\gamma}\right] \leq \frac{1}{2}.$$

If we simplify the last expression we obtain the two conditions

$$3\gamma \leq 1 \quad \text{and} \quad 3^{s+2} p \sqrt{s} (2\gamma\sqrt{s} q^{-1})^p \left[1 + \frac{p}{\gamma}\right] \leq \frac{1}{4}. \tag{6.28}$$

If we now use the assumptions that $L \geq 1/12$, and $p \geq s+1$, where $s \geq 2$, i.e. $p \geq 4$, since p is even, we easily see that

$$\frac{c^*Ac}{(2\pi)^s c_\gamma} \geq \frac{1}{2} \|c\|^2,$$

where c_γ is defined as in (6.9), from which the estimate (6.24) follows. This completes the proof. □

Corollary 6.7. *With the assumptions and notation of Theorem 6.6, we have that*

$$\|A^{-1}\| \leq \theta^{-1}. \tag{6.29}$$

Proof. As a consequence of (6.24), we obtain that the lowest eigenvalue of A is bounded below by $\theta = (2\pi)^s c_\gamma / 2$. If we now use the fact that $\|A^{-1}\|$ is the inverse of the lowest eigenvalue, we get inequality (6.29). □

Remark 6.8. The upper bound (6.29) depends only on the separation distance of the data sites, q, on the order of the B-spline, p, and on the space-dimension, s.

Note that one can easily eliminate the dependency on p by setting $p = s+1$ if s is odd, and $p = s+2$, if s is even. Hence, we obtain that the stability behavior of the matrix-valued functions is similar to the results presented in [22] for point evaluations and certain first order derivatives, and agrees with the results for point evaluations obtained in [21] for scalar-valued RBFs. Note also that the result can be generalized for arbitrary order-m $\mathcal{S}$-CPD matrix-valued RBFs by applying well-known techniques as those stated in [21, 25]. If only point evaluations or only partial derivatives are involved, the results of Theorem 6.6 and Corollary 6.7 can be adjusted easily by setting either all b_j's or all a_j's equal to zero, respectively. In both cases the resulting upper bounds remain unchanged.

And finally, we are now able to obtain the following explicit stability estimates for even and odd space dimension s.

Proposition 6.9. *For space dimension $s \geq 2$, let $\Phi(x)$ be given to have the form*

$$\Phi(x) = \int\limits_{\mathbb{R}^s} e^{ix\cdot\xi}\, d\mu(\xi),$$

with $d\mu(\xi)$ defined as in Theorem 6.6 and $\hat{\psi}$ being the positive Fourier transform of a scalar-valued RBF ψ such that Φ is an order-0 strictly $\mathcal{S}$-CPD function. We define $\varphi_0(r) := \min_{\|\xi\|\leq 2r} \hat{\psi}(\xi)$ and

$$\mathcal{V}_s := (s+1)\left[2\cdot 6^{s+2}(s+1)(s+2)s^{(s+2)/2}\right]^{1/s}, \qquad \mathcal{W}_s := 2\left[\frac{(s+1)B_{s+1}(0)}{\mathcal{V}_s}\right]^s$$

for odd space dimension $s \geq 3$, and

$$\mathcal{V}_s' := (s+2)\left[4\cdot 6^{s+2}(s+2)(s+3)s^{(s+3)/2}\right]^{1/(s+1)}, \qquad \mathcal{W}_s' := 2\left[\frac{(s+2)B_{s+2}(0)}{\mathcal{V}_s}\right]^s$$

for even space dimension $s \geq 2$. Then Φ has the following stability estimate:

$$\|A^{-1}\| \leq \mathcal{W}_s \frac{q^{s+1}}{\varphi_0\left(\frac{\mathcal{V}_s}{q^{(s+1)/s}}\right)} \tag{6.30}$$

in case of odd space dimension s, and

$$\|A^{-1}\| \leq \mathcal{W}'_s \frac{q^{s(s+2)/(s+1)}}{\varphi_0\left(\frac{\mathcal{V}'_s}{q^{(s+2)/(s+1)}}\right)} \tag{6.31}$$

in case of even space dimension s.

Proof. Observe that $\gamma^{s+1} \leq \gamma^s$ by the restrictions (6.23). In the case of odd space dimension, let $p = s+1$. If we now choose γ such that

$$\gamma^s = \frac{1}{2}\frac{q^{s+1}}{6^{s+2}(s+1)(s+2)s^{(s+2)/2}},$$

then (6.23) holds. Applying this γ to the inequality in (6.24), a short calculation yields that

$$\|A\| \geq \frac{1}{\mathcal{W}_s}\frac{\varphi_0\left(\frac{\mathcal{V}_s}{q^{(s+1)/s}}\right)}{q^{s+1}}.$$

Taking the reciprocal gives the desired result for odd space dimension. In the case of even space dimension, let $p = s+2$. If we choose γ such that

$$\gamma^{s+1} = \frac{1}{4}\frac{q^{s+2}}{6^{s+2}(s+2)(s+3)s^{(s+3)/2}},$$

then (6.23) holds. Applying this γ to the inequality in (6.24), a short calculation yields that

$$\|A\| \geq \frac{1}{\mathcal{W}'_s}\frac{\varphi_0\left(\frac{\mathcal{V}'_s}{q^{(s+2)/(s+1)}}\right)}{q^{s(s+2)/(s+1)}}.$$

Taking the reciprocal now gives the desired result, which completes the proof. $\square$

D. Examples

In this section we present some practical bounds obtained from investigating the Gaussian functions $\psi_\alpha(x) = e^{-\alpha\|x\|^2}$ for $\alpha > 0$, and the Wendland functions $\psi_{l,k}$ as defined in Chapter III. We only consider odd space dimensions, but of course similar results can be obtained for even space dimensions using the same techniques.

In order to obtain practical bounds for the Gaussian functions $\psi_\alpha(x) = e^{-\alpha\|x\|^2}$ for $\alpha > 0$, we need to estimate $\varphi_0\left(\frac{\mathcal{V}_s}{q^{(s+1)/s}}\right)$. The Fourier transform of ψ_α is given by $\hat{\psi}_\alpha(\xi) = (\pi/\alpha)^{s/2} e^{-\|\xi\|^2/(4\alpha)}$. Clearly, the function $\hat{\psi}_\alpha$ obtains its minimum on the boundary, i.e.

$$\inf_{\|\xi\|\le M} \hat{\psi}_\alpha(\xi) = (\pi/\alpha)^{s/2} e^{-M^2/(4\alpha)}. \tag{6.32}$$

We can now state the following corollary.

Corollary 6.10. *Let $\psi = \psi_\alpha$ be the Gaussian function with $\alpha > 0$ and let $A = A_\alpha$ be the corresponding interpolation matrix. Then the assumptions of Theorem 6.6 are fulfilled and for odd space dimension s we obtain*

$$\|A_\alpha^{-1}\| \le (\alpha/\pi)^{s/2} \mathcal{W}_s \, q^{s+1} e^{\frac{\mathcal{V}_s^2}{\alpha q^{2(s+1)/s}}}.$$

Proof. Using equation (6.32), we get that

$$\varphi_0\left(\frac{\mathcal{V}_s}{q^{(s+1)/s}}\right) = (\pi/\alpha)^{s/2} e^{-\frac{\mathcal{V}_s^2}{\alpha q^{2(s+1)/s}}}.$$

Combining this equation and inequality (6.30) leads to the result. □

We now finish this chapter with an investigation of stability estimates involving Wendland functions $\psi_{l,k}$. Wendland [29, Theorem 3.6] proved that the Fourier transform of $\psi_{l,k}$ possesses the following lower bound:

$$\hat{\psi}_{l,k}(\xi) \geq c_1 \|\xi\|^{-(s+2k+1)} \text{ for } \|\xi\| \geq r_0 \tag{6.33}$$

with constants $c_1 > 0$, $r_0 \geq 0$. Therefore, we obtain the following statement:

Corollary 6.11. *Let $\psi = \psi_{l,k}$ be a Wendland function in $C^{2k}(\mathbb{R}^s)$ with $l > 1$ such that $l = \lfloor s/2 \rfloor + k + 1$. Let $A = A_{l,k}$ be the corresponding interpolation matrix. Then the assumptions of Theorem 6.6 are fulfilled and for odd space dimension s we obtain*

$$\|A_{l,k}^{-1}\| \leq \mathcal{O}\left(q^{-(2k+1)\frac{s+1}{s}}\right), \text{ as } q \to 0.$$

Proof. If we use (6.33), we then get

$$\varphi_0\left(\frac{\mathcal{V}_s}{q^{(s+1)/s}}\right) \geq C(s,k) q^{(s+1)+(2k+1)\frac{s+1}{s}}.$$

Inequality (6.30) then gives

$$\|A_{l,k}^{-1}\| \leq \mathcal{W}_s \frac{q^{s+1}}{\varphi_0\left(\frac{\mathcal{V}_s}{q^{(s+1)/s}}\right)} \leq C'(s,k) q^{-(2k+1)\frac{(s+1)}{s}},$$

which yields the result. □

In the explicit case $\psi_{l,k} = \psi_{3,1} \in C^2$ for $s = 2, 3$, then the matrix-valued function $\Phi_{3,1}$ is in $C(\mathbb{R}^s)$. Therefore, we only consider point evaluations, i.e. $b_j = 0$ for all $1 \leq j \leq N$. Wendland showed in [28, Proposition 6.25] that the Fourier transform of $\psi_{3,1}$ has the lower bound

$$\hat{\psi}_{3,1}(\xi) \geq \frac{4}{5\sqrt{2\pi}} \|\xi\|^{-6} \tag{6.34}$$

for $\|\xi\| \geq \sqrt{26}$, which is not a major restriction as discussed in [28]. Hence, for the matrix-valued basis function involving the function $\psi_{3,1}$ we obtain the following stability estimate which concludes this chapter.

Corollary 6.12. *Let $\psi = \psi_{3,1}$ be the C^2 Wendland function with the space dimension $s = 2, 3$. Let $A = A_{3,1}$ be the corresponding interpolation matrix. Then the assumptions of Theorem 6.6 are fulfilled and for odd space dimension s we obtain*

$$\|A_{3,1}^{-1}\| \leq \mathcal{W}_s \frac{5\sqrt{2\pi}M}{4} q^{s+1} \text{ for } M \geq \sqrt{26}.$$

CHAPTER VII

APPLICATIONS

A. Description of the Problem

In this chapter we investigate applications involving divergence-free radial basis functions based on the two-dimensional *driven cavity problem*. Consider a rectangular-shaped cavity with horizontal fluid flow on its top, B1, and zero flow on its three remaining sides, B2, B2, and B4, as described in Figure 3. The task is to model the flow in the inside of the cavity after it reaches its steady–state condition. We used MATLAB to obtain visualizations of the results. The program can be found in Appendix B.

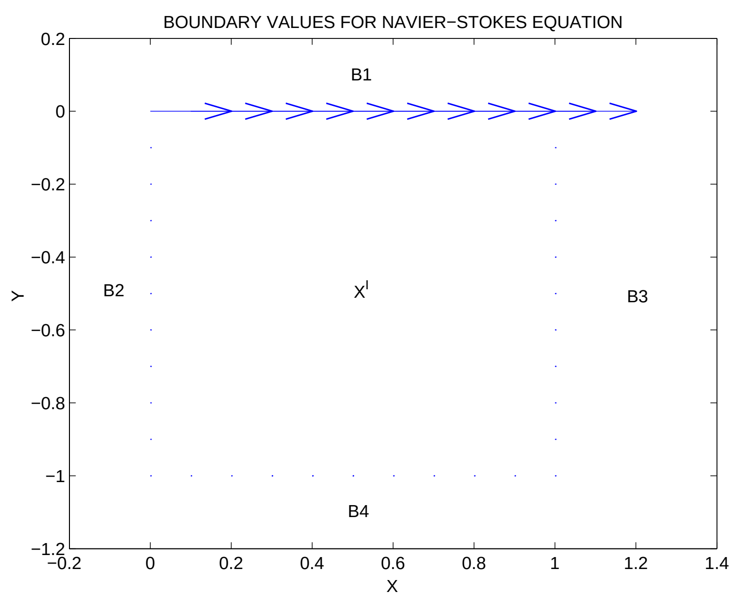

FIGURE 3. The cavity with the initial boundary conditions

We assume that the flow is *incompressible*, i.e. it has constant density. The differential equation that describes an incompressible fluid flow is the Navier-Stokes equation. The fluid we consider is air. We would like to obtain a numerical solution to the driven cavity problem that describes the velocity $\mathbf{u}$ of the Navier-Stokes equation of incompressible fluid flow, i.e. it fulfills the system

$$\begin{cases} \dfrac{D\mathbf{u}}{Dt} = -\nabla p + \nu \Delta \mathbf{u}, \\ \qquad \nabla \cdot \mathbf{u} = 0 \end{cases} \tag{7.1}$$

in the interior, X^I, of the cavity, where

$$\frac{D\mathbf{u}}{Dt} := \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}$$

is the *material derivative*, p is the *pressure*, and ν is the *viscosity coefficient*.

Note that the second equation in (7.1) means that the flow is divergence-free. We therefore choose an interpolation function which is based on a divergence-free matrix-valued RBF. We use strictly positive definite and compactly-supported functions, since this leads to a sparse, symmetric, and positive definite interpolation matrix and hence speeds up the algorithm. In order to guarantee the uniqueness of a solution we need boundary conditions. They are given as

$$\begin{cases} \mathbf{u} = (1,0)^T \quad \text{on B1}, \\ \mathbf{u} = (0,0)^T \quad \text{on B2, B3, B4}, \end{cases} \tag{7.2}$$

which means that we have a horizontal air flow on top of the cavity, and zero air flow on the remaining three boundaries.

B. Constant Pressure

We first consider a simplified version of the Navier-Stokes equation in two dimensions. We make the following two assumptions:

A1 The fluid flow problem is time-independent, and

A2 The fluid flow has constant pressure.

The assumptions **A1** and **A2** imply the equations $\frac{\partial \mathbf{u}}{\partial t} = 0$ and $\nabla p = 0$. A mixed formulation of the incompressible Navier-Stokes equation leads to a vanishing pressure term, which suggests a reasonable motivation for collocation methods applied to the pressure-free case.

Under the assumptions **A1** and **A2**, the problem (7.1) can be restated as follows. Find $\mathbf{u} = (u, v)^T$ such that

$$
\begin{cases} \nu \Delta \mathbf{u} = (\mathbf{u} \cdot \nabla)\mathbf{u} & \text{on } X^I, \\ \quad \mathbf{u} = (1, 0)^T & \text{on B1}, \\ \quad \mathbf{u} = (0, 0)^T & \text{on B2, B3, B4}, \end{cases} \tag{7.3}
$$

where ν is the viscosity coefficient. Note that in (7.3), we do not need to require

$$
\nabla \cdot \mathbf{u} = 0,
$$

since our interpolant fulfills the condition naturally.

The system (7.3) is non-linear based on the term $(\mathbf{u} \cdot \nabla)\mathbf{u}$ in the first equation. Therefore, we use an iterative method to solve the driven cavity problem numerically. The numerical method is as follows. We first construct an initial interpolant $\mathbf{s}_0$ which interpolates the fluid flow only on the four boundaries. Next, we define an interpolant $\mathbf{s}_r$ which interpolates the fluid flow on the boundaries, as well as in the interior. As interpolation points we chose a regular grid on the cavity. Since the data on the boundary stems from plain point evaluations, we reflect this in our interpolant $\mathbf{s}_0$. In the interior we not only have data stemming from point evaluations, but also derivative information, which is reflected in the interpolant $\mathbf{s}_r$ as well. We now consider the different interpolation problems arising from solving (7.3) in more detail.

The initial interpolation of the boundary values is stated as follows: Take an interpolant of form

$$\mathbf{s}_0(x) = \sum_{k=1}^{N^B} \alpha^B_{0,k} \Phi(x - x^B_k),$$

find α_0 such that

$$\mathbf{s}_0(x^B_l) = \mathbf{d}^B_l \text{ for } x^B_l \in X^B$$

for boundary data $D^B = \{\mathbf{d}^B_l\}_{l=1}^{N^B}$ at the boundaries sites $X^B = \{x^B_l\}_{l=1}^{N^B}$ of the cavity as stated in (7.3). Here, B is the union of all four boundaries, and N^B is the number of boundary interpolation points. Note that the interpolant $\mathbf{s}_0$ only consists of terms of translates of radial basis functions. Hence, it can be rewritten as a linear combination of convolutions of delta functionals with radial basis functions, i.e.

$$\mathbf{s}_0(x) = \sum_{k=1}^{N^B} \alpha^B_{0,k} (\delta_{x^B_k} * \Phi)(x),$$

since the initial interpolation problem only consists of data arising from point evaluations. The initial interpolation on the boundary points is pictured in Figure 4. As expected, it looks very similar to the original data shown in Figure 3. We made the flow on the boundary sides B1, B2, and B3 slightly larger than zero, i.e. we assigned the values $\mathbf{u} = (0.01, 0)^T$, in order to make the results more visible.

The iterative interpolation of the Navier–Stokes equation can be described as follows:

For r=1:nsteps

Take interpolant

$$\mathbf{s}_r(x) = \sum_{i=1}^{N^I} \alpha^I_{r,i} \Delta\Phi(x - x^I_i) + \sum_{k=1}^{N^B} \alpha^B_{r,k} \Phi(x - x^B_k),$$

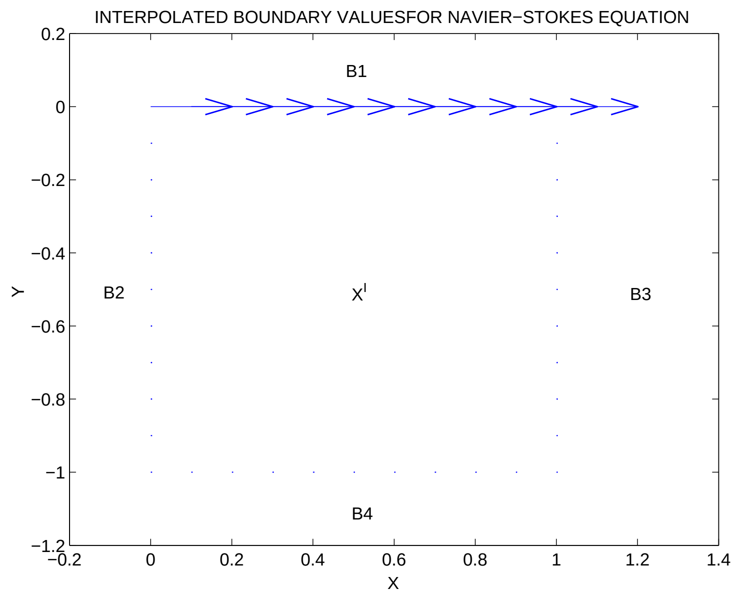

FIGURE 4. Interpolated boundary values for the Navier-Stokes equation

find coefficient vector α_r such that

$$\begin{cases} (\nu\Delta - (\mathbf{s}_{r-1}(x_j^I) \cdot \nabla))\mathbf{s}_r(x_j^I) = \mathbf{0} & \text{for } x_j^I \in X^I, \\ \mathbf{s}_r(x_l^B) = \mathbf{d}_l^B & \text{for } x_l^B \in X^B, \end{cases}$$

where $\mathbf{d}_l^B$ is given by (7.3).

end

Note that the iterative interpolant consists of terms that stem from the Laplacian operator applied to the radial basis functions and of terms arising from translates of radial basis functions. Therefore the interpolant can be rewritten as

$$\mathbf{s}_r(x) = \sum_{i=1}^{N^I} \alpha_{r,i}^I (\Delta\delta_{x_i^I} * \Phi)(x) + \sum_{k=1}^{N^B} \alpha_{r,k}^B (\delta_{x_k^B} * \Phi)(x).$$

The reason for the choice of the interpolant is that in the iterative interpolation problem we interpolate data stemming from point evaluations on the boundary, as well as Laplacian data on the interior of the cavity. The iteration is performed until it reaches its steady state, i.e. until the difference of two consecutive interpolants becomes very small.

1. Setup and Results

In this section we describe the setup of the problem which we wrote an algorithm for and give its results. The divergence-free RBF is based on the two-dimensional scalar-valued Wendland function $\psi_{6,4}(r) \doteq (1-r)_+^{10}(85.8r^4+90r^3+42r^2+10r+1) \in C^8$ shown in Figure 5. We choose this function since the problem (7.3) involves second-order

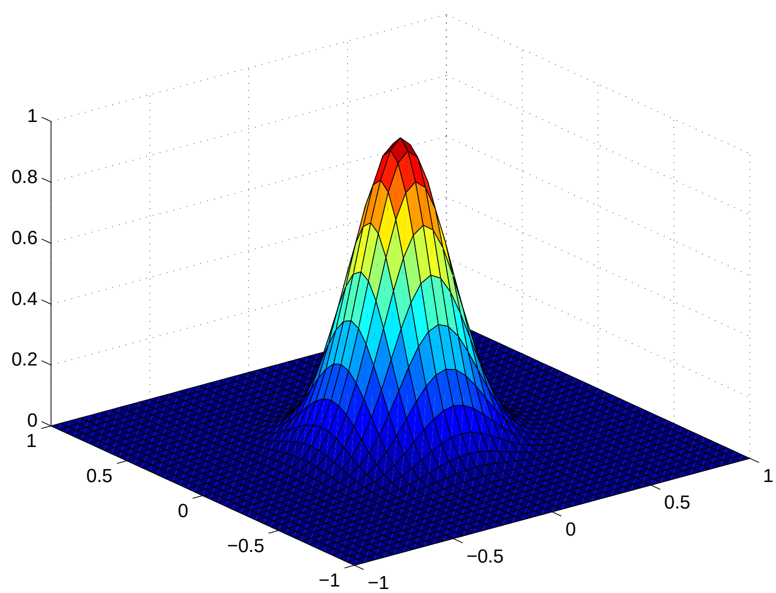

FIGURE 5. $\psi_{6,4}(r) \doteq (1-r)_+^{10}(85.8r^4+90r^3+42r^2+10r+1)$

derivatives, while the interpolant is also based on second-order derivative information. Finally, the construction of the divergence-free matrix-valued RBF requires second-order derivatives as well. Therefore, we need at least a C^6 generating function. We choose a C^8 function to obtain smoother results. The resulting divergence-free matrix-valued RBF $\Phi = \Phi_{6,4}$ is of the form

$$\Phi = \begin{pmatrix} \Phi_{1,1} & \Phi_{1,2} \\ \Phi_{2,1} & \Phi_{2,2} \end{pmatrix}, \tag{7.4}$$

with

$$\begin{aligned}
\Phi_{1,1} &= \frac{26}{5}(1-r)^8(5+40r+114x^2-18y^2+24r(3x^2-41y^2)-231r^2(x^2+13y^2)), \\
\Phi_{1,2} &= \Phi_{2,1} = \frac{3432}{4}(1-r)^8(1+8r+21r^2)xy, \\
\Phi_{2,2} &= \frac{26}{5}(1-r)^8(5+40r+114y^2-18x^2+24r(3y^2-41x^2)-231r^2(13x^2+y^2)),
\end{aligned}$$

if $r = \sqrt{x^2+y^2} \leq 1$, and zero otherwise. In order to obtain the interpolants $\mathbf{s}_0$ and $\mathbf{s}_r$ we derive all necessary derivatives analytically. This has the advantage that we get higher accuracy for the results than by applying numerical differentiation schemes in order to obtain the necessary derivatives.

The setup is as follows. Given is a rectangular grid with edges $(0,0)$, $(l,0)$, $(0,-d)$, and $(l,-d)$, where $l, d > 0$. The number of centers is $N = 121$, i.e. 11 points in each direction (which we also express by the notation $n = 10$). The viscosity coefficient is set to be $\nu = 0.1$, and the number of iterations is $nsteps = 15$.

We start with investigating a square-shaped cavity with side length 1. We therefore set $l = d = 1$. We first give a visualization of the result obtained by the program and then analyze important errors. The calculated steady-state solution to (7.3) is shown in Figure 6.

We obtain the following errors: The l_2–error of the initial interpolation is

$$\|\mathbf{u} - \mathbf{s}_0\|_2 = 1.1811e - 014. \tag{7.5}$$

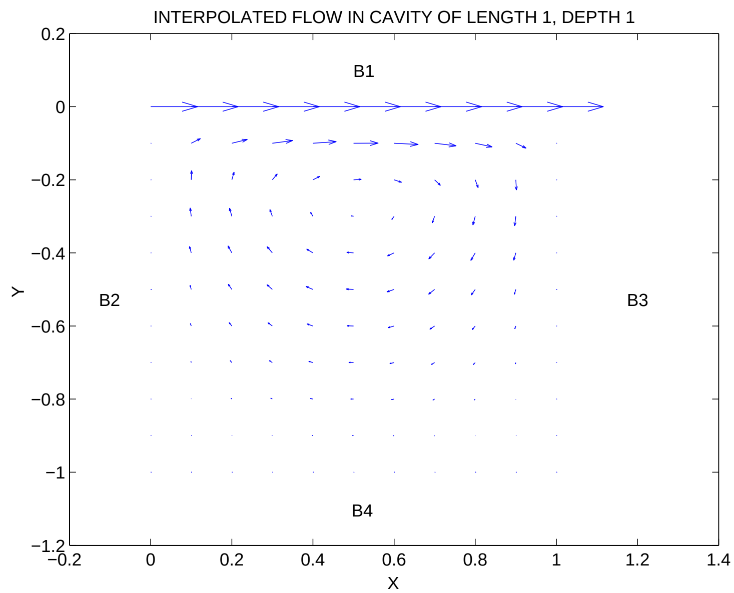

FIGURE 6. Interpolated Navier-Stokes equation on a square cavity

We now define certain errors of interest:

$$\begin{aligned}
\text{error}_1 &:= \|\mathbf{s}_r - \mathbf{s}_{r-1}\|_2, \\
\text{error}_2 &:= \|\mathbf{s}_r - \mathbf{s}_{r-1}\|_2 + \|\nabla \cdot (\mathbf{s}_r - \mathbf{s}_{r-1})\|_2, \\
\text{error}_3 &:= \frac{\|\mathbf{s}_r - \mathbf{s}_{r-1}\|_2 + \|\nabla \cdot (\mathbf{s}_r - \mathbf{s}_{r-1})\|_2}{\|\mathbf{s}_{r-1} - \mathbf{s}_{r-2}\|_2 + \|\nabla \cdot (\mathbf{s}_{r-1} - \mathbf{s}_{r-2})\|_2}, \\
\text{error}_4 &:= \|\, LHS_r - RHS_r\|_2,
\end{aligned}$$

where LHS and RHS stands for left hand side and right hand side, respectively. The errors are chosen such that they reflect convergence of the solution. These specific l_2-errors of the iterative interpolation are given in Table 2. A visualization of these errors is given in Figure 7.

From (7.5) we observe that the initial interpolation is very good, since its error is of order $\mathcal{O}(10^{-14})$. Because the step size, h, is given as $h = l/n$, where $l = 1$ and $n = 10$, we have a step size of $h = 0.1$ in each direction and hence the initial

TABLE 2. l_2-errors of the iterative interpolation for $l = d = 1$

iterat.	error_1	error_2	error_3	error_4
1	1.5163	23.4318	–	0.9403
2	0.0348	0.5324	44.0119	0.0304
3	0.0016	0.0171	31.1663	0.0014
4	5.7567e-005	7.5191e-004	22.7187	5.7793e-005
5	2.7795e-006	3.0753e-005	24.4498	2.5560e-006
6	1.0420e-007	1.4066e-006	21.8629	1.0406e-007
7	5.0941e-009	5.6264e-008	25.0008	4.6547e-009
8	1.9080e-010	2.5713e-009	21.8815	1.9048e-010
9	9.3598e-012	1.0271e-010	25.0349	8.3100e-012
10	3.4972e-013	5.4551e-012	18.8282	8.3822e-013
11	1.3522e-013	2.6849e-012	2.0317	1.1001e-012
12	1.6620e-013	3.8632e-012	0.6950	8.8051e-013
13	1.6744e-013	4.1007e-012	0.9421	1.0493e-012
14	1.3889e-013	3.1925e-012	1.2845	9.1174e-013
15	1.4877e-013	3.6060e-012	0.8853	9.4192e-013

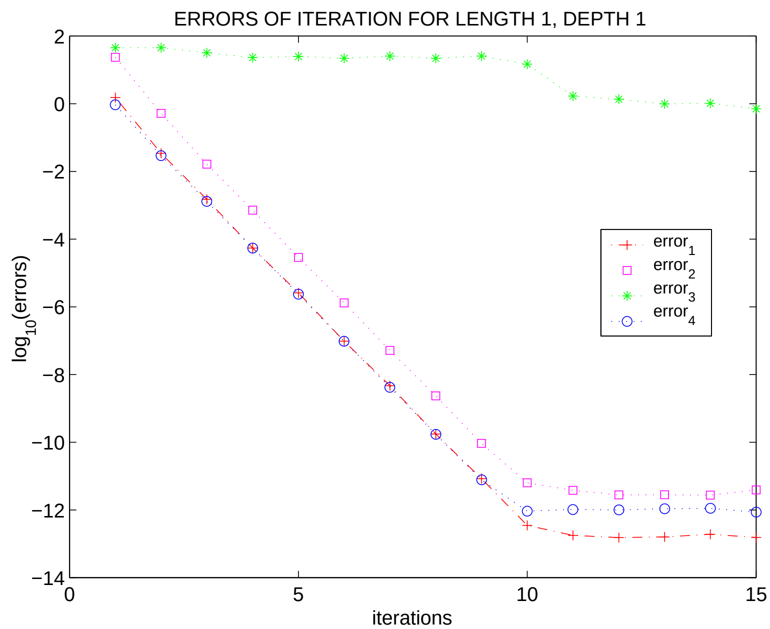

FIGURE 7. Visualization of specific errors, $l = d = 1$

l_2-error is of very high order. The interpolation errors obtained for the iterative solutions are almost of the same order, i.e. they are of order $\mathcal{O}(10^{-13})$, which holds for error_1 and error_4. The second error, error_2, is one order higher, since it contains first-order derivative information. As expected, error_3 is about 1, which reflects that the numerical solution s_r almost reaches its steady-state.

In order to get an estimate for the order of approximation we assume that the interpolant behaves like $\|\mathbf{u}-\mathbf{s}_r\| \leq \|\mathbf{u}\|Ch^r$ for some integer r, where h is the coefficient of grid spacing. Then we can define

$$\text{norm}_{\text{A}} := \frac{\|\mathbf{s}_{r,2h} - \mathbf{s}_{r,h}\|}{\|\mathbf{s}_{r,4h} - \mathbf{s}_{r,2h}\|} \approx \left(\frac{1}{2}\right)^r.$$

Results for the norm of the application and r for start grids with $3, 6, 7$, and 10 points in each direction are given in Table 3.

As we can see, the order of approximation is between 1.4 and 2.4, depending on

TABLE 3. Values for approximation order r

n	$l = d$	norm_A	$r = log_{0.5}(\text{norm}_A)$
2	1	0.2807	1.8329
5	2	0.3767	1.4085
6	3	0.3613	1.4687
9	4	0.1854	2.4313

how many interpolation points we choose, as well as on the size of the cavity. This is related to the fact that we chose functions with fixed local support of length 1. Therefore, if we keep the ratio of the support of the RBF and the step size , $h = l/n$, constant, then we obtain similar results for the order of approximation, r.

2. Other Shapes Than Square

We now investigate cavities with other shapes. We first consider a cavity that is less deep than long, i.e. it has a depth of $d = 0.6$ and a side length of $l = 1$. The resulting interpolated fluid flow is given in Figure 8.

As expected, the air flow is put in motion right on the top of the cavity, and moves in almost horizontal flow of small depth over the cavity. On its boundaries the flow now makes a circular move inside the cavity and is pushed up on the other side. As expected, we obtain a flow that reaches deeper into the cavity than in the square case. The horizontal flow on top of the cavity is strong enough to reach down to its bottom. This is physically appropriate and hence a nice model of the incompressible airflow. The solution is divergence-free as well, by its construction.

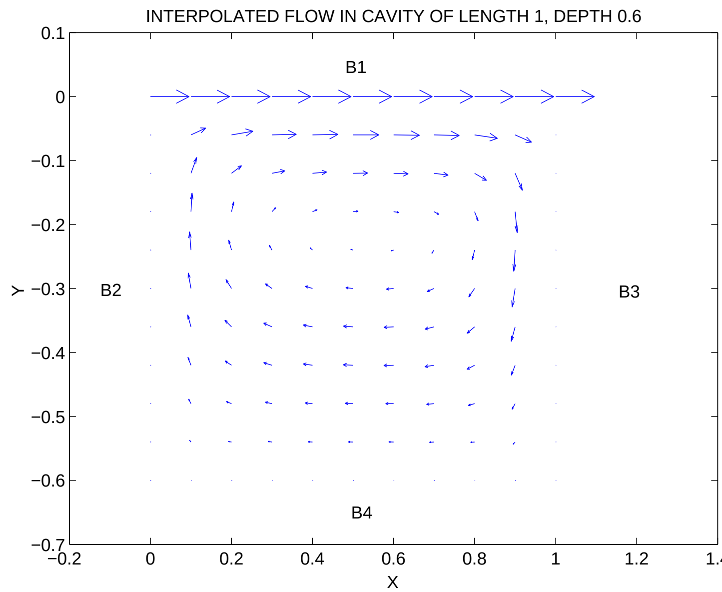

FIGURE 8. Interpolated Navier-Stokes equation on a shallow cavity

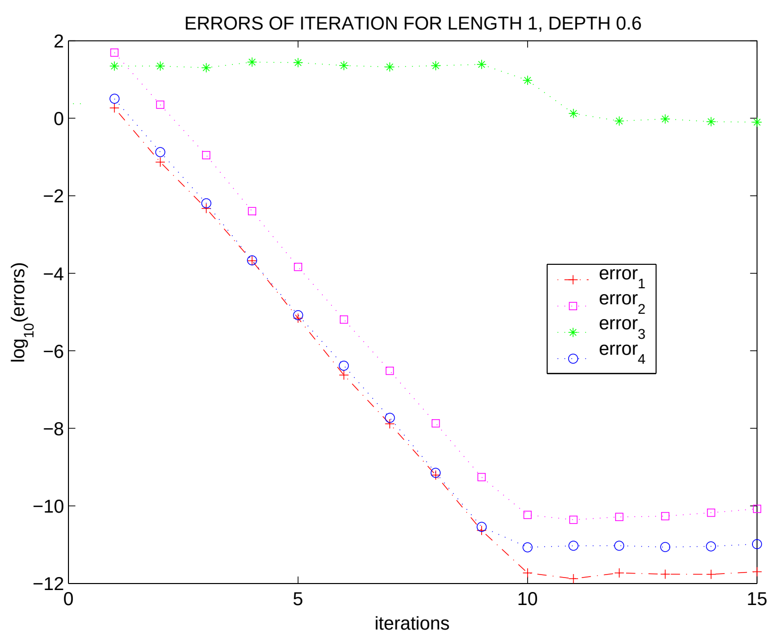

FIGURE 9. Visualization of specific errors, $l = 1$, $d = 0.6$

We obtain the following errors: The l_2–error of the initial interpolation is

$$\|\mathbf{u} - \mathbf{s}_0\|_2 = 8.3202e - 014.$$

The l_2-errors of the iterative interpolation are as given in Table 4. A visualization of these errors is given in Figure 9.

TABLE 4. l_2-errors of the iterative interpolation for $l = 1, d = 0.6$

iterat.	error_1	error_2	error_3	error_4
1	1.8635	49.8548	–	3.2322
2	0.0748	2.2816	21.8511	0.1382
3	0.0049	0.1144	19.9516	0.0066
4	2.1666e-004	0.0041	27.8098	2.2472e-004
5	7.2263e-006	1.5290e-004	26.8932	8.8178e-006
6	2.5122e-007	6.8104e-006	22.4512	4.4177e-007
7	1.4016e-008	3.2761e-007	20.7881	2.0284e-008
8	6.7579e-010	1.4602e-008	22.4368	7.7972e-010
9	2.5378e-011	6.1345e-010	23.8022	3.0605e-011
10	2.1058e-012	9.3693e-011	6.5475	9.9502e-012
11	2.1331e-012	9.9775e-011	0.9390	1.0556e-011
12	1.4783e-012	6.3603e-011	1.5687	1.0276e-011
13	1.7440e-012	6.5963e-011	0.9642	1.0166e-011
14	1.8888e-012	7.2626e-011	0.9083	8.9400e-012
15	2.3050e-012	7.2357e-011	1.0037	8.4095e-012

We observe that the errors are almost as good as in the case of a square cavity. The loss of one order might result from the fact that the cavity is no longer symmetric concerning its length and depth.

We conclude this section by investigating a cavity that is less long than deep, i.e. it has a side length of $l = 0.6$ and a depth of $d = 1$. The resulting interpolated fluid flow is given in Figure 10. As expected, we obtain a flow that barely goes into

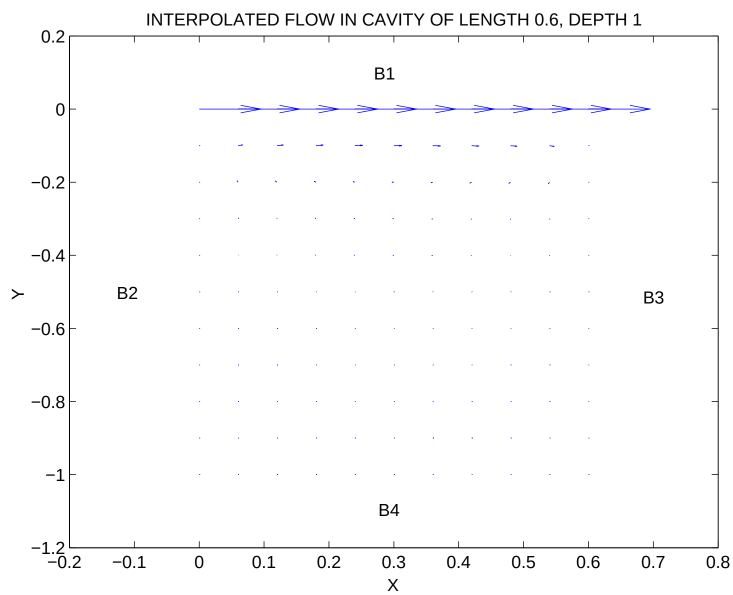

FIGURE 10. Interpolated Navier-Stokes equation on a deep cavity

depth of the cavity. Since the surface of the top is now smaller than the length of the cavity, the air does not reach very deeply into the cavity. We obtain the following errors: The l_2–error of the initial interpolation is

$$\|\mathbf{u} - \mathbf{s}_0\|_2 = 1.3006e - 014.$$

The l_2-errors of the iterative interpolation are given in Table 5. A visualization of these errors is given in Figure 11.

TABLE 5. l_2-errors of the iterative interpolation for $l = 0.6, d = 1$

iterat.	error_1	error_2	error_3	error_4
1	0.6750	33.0436	–	0.6657
2	0.0060	0.1566	210.9673	0.0039
3	3.5909e-005	0.0014	115.9604	3.1525e-005
4	2.4918e-007	9.1837e-006	147.0763	2.2035e-007
5	2.3152e-009	4.8094e-008	190.9559	1.9427e-009
6	1.8635e-011	5.3372e-010	90.1106	2.0424e-011
7	4.9977e-013	1.2046e-011	44.3048	2.4432e-012
8	4.1381e-013	1.0518e-011	1.1453	2.5613e-012
9	3.8566e-013	1.1649e-011	0.9029	2.2021e-012
10	3.7408e-013	1.0826e-011	1.0761	2.5367e-012
11	3.9358e-013	9.9858e-012	1.0841	2.3099e-012
12	3.8868e-013	1.0580e-011	0.9438	2.1498e-012
13	3.3144e-013	9.6894e-012	1.0920	2.1947e-012
14	3.9638e-013	1.1208e-011	0.8645	2.1957e-012
15	4.1154e-013	1.0331e-011	1.0849	2.6723e-012

We observe that the errors of the iterative interpolant behaves similarly to the previous case.

C. General Pressure

We conclude this chapter with an investigation of a more general form of the Navier-Stokes equation. We derive an interpolation method for the case that the pressure is not constant. We still assume that the problem is time-independent. We therefore

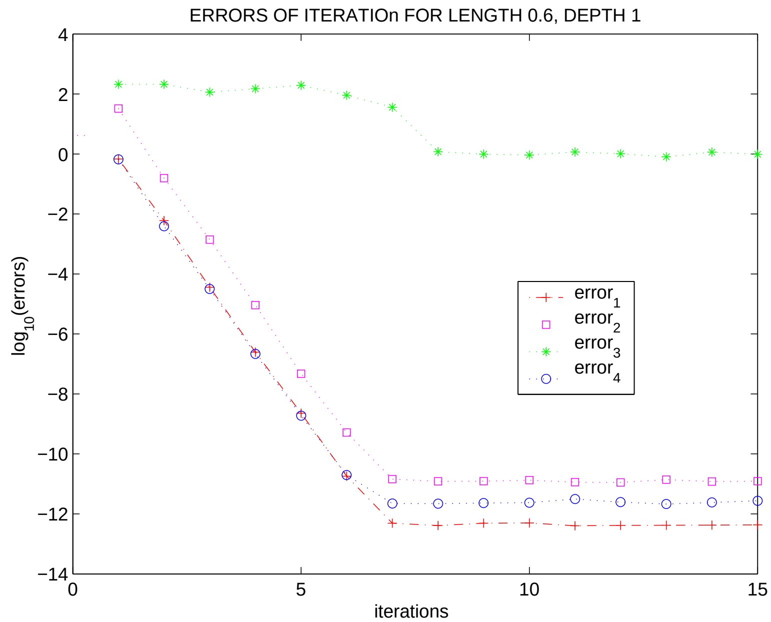

FIGURE 11. Visualization of specific errors, $l = 0.6$, $d = 1$

only have one assumption,

A1 The fluid flow problem is time–independent,

i.e. we release the assumption of constant pressure of the previous section. The problem arising from the Navier-Stokes equation is now as follows: Find $\mathbf{u} = (u, v)^T$ such that

$$(7.6) \qquad \begin{cases} (\nu\Delta - (\mathbf{u}\cdot\nabla))\mathbf{u} = \nabla p & \text{on } X^I, \\ \mathbf{u} = (1,0)^T & \text{on B1}, \\ \mathbf{u} = (0,0)^T & \text{on B2, B3, B4.} \end{cases}$$

1. Derivation of a Numerical Method

Applying the gradient to the first equation of (7.6 yields

$$
\begin{aligned}
\Delta p &= \nabla \cdot [\nu \Delta \mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{u}] \\
&= \underbrace{\nabla \cdot [\nu \Delta \mathbf{u}]}_{0} - \nabla \cdot [(\mathbf{u} \cdot \nabla)\mathbf{u}] \\
&= -\nabla \cdot [(\mathbf{u} \cdot \nabla)\mathbf{u}]
\end{aligned} \tag{7.7}
$$

on the interior of the cavity. The first term vanishes because of the fact that $\mathbf{u}$ is divergence free. In order to obtain a unique solution for the pressure, we derive boundary conditions. Note that the first equation of (7.6) is equivalent to

$$
\begin{pmatrix} \partial_1 p \\ \partial_2 p \end{pmatrix} = \begin{pmatrix} \nu(\partial_1^2 + \partial_2^2)u - (u\partial_1 u + v\partial_2 u) \\ \nu(\partial_1^2 + \partial_2^2)v - (u\partial_1 v + v\partial_2 v) \end{pmatrix} \tag{7.8}
$$

in two dimensions, where $\mathbf{u} = (u, v)^T$. Hence, the *Neumann conditions* on the four boundaries are given by

$$
\begin{cases}
\partial_2 p = \nu \partial_2^2 v & \text{on B1,} \\
\partial_1 p = \nu \partial_1^2 u & \text{on B2,} \\
\partial_1 p = -\nu \partial_1^2 u & \text{on B3,} \\
\partial_2 p = -\nu \partial_2^2 v & \text{on B4,}
\end{cases} \tag{7.9}
$$

based on the boundary values expressed in (7.6). Therefore, the numerical method that solves for the pressure p is given as follows: Given $\mathbf{s}_{r-1}$ and $\mathbf{s}_r = (u_r, v_r)^T$, find p such that

$$
\begin{cases}
(\nu\Delta - (\mathbf{s}_{r-1} \cdot \nabla))\mathbf{s}_r = \nabla p & \text{on } X^I, \\
\partial_2 p = \pm \nu \partial_2^2 v_r & \text{on B1, B4,} \\
\partial_1 p = \pm \nu \partial_1^2 u_r & \text{on B2, B3.}
\end{cases} \tag{7.10}
$$

Since partial-derivative operators, as well as the Laplacian operator are both applied to the pressure, the interpolation data is also based on this information. Therefore, it is natural to construct the interpolant for the pressure involving these operators as well. We define the scalar-valued interpolant for the pressure to be

$$p_r(x) := \sum_{s=1}^{N^h} \beta_{r,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{r,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{r,k}^I \Delta g(x - x_k^I), \tag{7.11}$$

where $x^h \in$ B1$\cup$B4 and $x^v \in$ B2$\cup$B3 stands for the horizontal and vertical boundaries, respectively.

2. An Algorithm

We now conclude this chapter by describing the numerical method which we derived in order to simultaneously solve for the pressure and the fluid flow in the Navier-Stokes equation. It is based on two iteration loops, the first one to obtain a first solution for the fluid flow that is based on constant pressure, and a second loop that solves for the pressure, followed by an interpolation of the fluid flow involving this pressure function, until the second loop reaches its steady-state. The method has not been implemented yet, but is of interest for future research. Note that the implementation of the pressure in the interpolation method results in taking two more derivatives, hence the choice of the scalar-valued C^8 Wendland RBF is appropriate. The algorithm completes this chapter.

i) Find the coefficient vector α_0 such that

$$\mathbf{s}_0(x) = \sum_{k=1}^{N^B} \alpha_{0,k}^B \Phi(x - x_k^B)$$

solves

$$\mathbf{s}_0(x_l^B) = \mathbf{d}_l^B \text{ for } x_l^B \in X^B$$

for boundary data $D^B = \{\mathbf{d}_l^B\}_{l=1}^{N^B}$ as in the system (7.3)

ii) **For r=1:nsteps**

Find the coefficient vector α_r such that

$$\mathbf{s}_r(x) = \sum_{i=1}^{N^I} \alpha_{r,i}^I \Delta\Phi(x - x_i^I) + \sum_{k=1}^{N^B} \alpha_{r,k}^B \Phi(x - x_k^B)$$

solves

$$\begin{cases} (\nu\Delta - (\mathbf{s}_{r-1}(x_j^I) \cdot \nabla))\mathbf{s}_r(x_j^I) = \mathbf{0} & \text{for } x_j^I \in X^I, \\ \mathbf{s}_r(x_l^B) = \mathbf{d}_l^B & \text{for } x_l^B \in X^B, \end{cases}$$

where $\mathbf{d}_l^B$ is given by (7.3).

end

iii) Find the coefficient vector β_r such that

$$p(x) := \sum_{s=1}^{N^h} \beta_{m,s}^h \partial_2 g(x - x_s^h) + \sum_{t=1}^{N^v} \beta_{m,t}^v \partial_1 g(x - x_t^v) + \sum_{k=1}^{N^I} \beta_{m,k}^I \Delta g(x - x_k^I)$$

solves

$$\begin{cases} \nabla p(x_j^I) = (\nu\Delta - (\mathbf{s}_{nsteps-1}(x_j^I) \cdot \nabla))\mathbf{s}_{nsteps}(x_j^I) & \text{for } x_j^I \in X^I, \\ \partial_2 p(x_l^h) = \pm\nu\partial_2^2 v_{nsteps}(x_l^h) & \text{for } x_l^h \in \text{B1, B4}, \\ \partial_1 p(x_i^v) = \pm\nu\partial_1^2 u_{nsteps}(x_i^v) & \text{for } x_i^v \in \text{B2, B3}. \end{cases}$$

iv) **For r=1:nsteps**

Find the coefficient vector α_r such that $\mathbf{s}_r(x)$ defined as in step ii) solves

$$\begin{cases} (\nu\Delta - (\mathbf{s}_{r-1}(x_j^I)\cdot\nabla))\mathbf{s}_r(x_j^I) = \nabla p(x_j^I) & \text{for } x_j^I \in X^I, \\ \mathbf{s}_r(x_l^B) = \mathbf{d}_l^B & \text{for } x_l^B \in X^B, \end{cases}$$

where $\mathbf{d}_l^B$ is given by (7.3).

end

v) Repeat steps iii) and iv) until a steady-state solution for the fluid flow, as well as for the pressure, is obtained.

CHAPTER VIII

SUMMARY AND FUTURE DIRECTIONS

We introduced a new class of matrix-valued radial basis functions that are divergence free as well as of compact support. Firstly, we proved several properties of these new functions, i.e. that they are divergence free, have compact support, and are positive definite. Secondly, we obtained a density result that guarantees that any divergence-free function f can be approximated arbitrarily well by a linear combination of divergence-free radial basis functions. This justified further studies of approximation and interpolation methods that are based on our new class of functions. We then derived error bounds for a generalized Hermite interpolation problem based on this new class of functions. The estimates obtained for the matrix-valued divergence-free RBFs of compact support are comparable to those obtained in [23] for scalar-valued RBFs of compact support. The estimates are given in terms of the mesh norm, as well as of the data-generating function, and they are independent of the number of points, N, as desired. Next, we derived upper bounds for stability estimates which are similar to the results obtained in [22] involving divergence-free RBFs based on the Gaussian functions. These upper bounds obtained are given in terms of the separation distance, q, and of the space dimension, s, and they are independent of the number of centers, N, as desired. We developed an algorithm to numerically solve PDEs arising from the driven cavity problem. These PDEs are based on the Navier-Stokes equation for incompressible fluid flows. Our results obtained are physically reasonable.

There are several directions we can take in extending and applying our results. Firstly, it seems of interest to extend the error estimates so that they are based on interpolation problems involving a larger class of radial basis functions instead of strictly positive definite functions, as for example conditionally positive definite matrix-valued RBFs or functions escaping the native space. This would offer the potential user a larger class of functions with additional properties. The same holds for the stability estimates as well.

The theory developed in [22] is not limited to divergence-free functions. Functions with other physical properties, as for example curl-free functions, can also be constructed, and it would be of interest to investigate their behavior. Moreover, further research can be done by developing functions that possess other physical properties frequently observed in experiments. This would be of help if one is interested in reflecting the properties of the experiment in the numerical solution, as for example convexity or positivity of the function.

One other future project of interest is to compare the results obtained for the Navier-Stokes equation using divergence-free matrix-valued RBFs to results obtained employing methods such as finite elements, or other numerical approaches. The implementation of an interpolant of the pressure in the Navier-Stokes equation is also a possible research direction. The study of equations arising from applications other than the Navier-Stokes equations might be of interest, especially equations involving industrial applications.

REFERENCES

[1] S. Agmon, *Elliptic Boundary Value Problems*, D. Van Nostrand Company, Inc., New York, 1965, pp. 11–14.

[2] M. D. Buhmann, *New developments in the theory of radial basis function interpolation*, Multivariate Approximation: From CAGD to Wavelets (K. Jetter and F. I. Uteras, eds.), World Scientific, Singapore, 1993, pp. 35–75.

[3] C. de Boor, *A Practical Guide to Splines*, Springer Verlag, New York, 1978.

[4] J. Duchon, *Interpolation des fonctions de deux variables suivant le principle de la flexion des plaques minces*, RAIRO Analyse Numerique **10** (1976), 5–12.

[5] ______, *Sur l'erreur d'interpolation des fonctions de plusiers variables par les d^m-splines*, RAIRO Analyse Numerique **12** (1978), 325–334.

[6] ______, *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, Constructive Theory of Functions of Several Variables (W. Schempp and K. Zeller, eds.), Springer Verlag, Berlin, 1977, pp. 85–100.

[7] N. Dyn, *Interpolation and approximation by radial and related functions*, Approximation Theory VI (L.L. Schumaker C. K. Chui and J. D. Ward, eds.), Academic Press, New York, 1989, pp. 211–234.

[8] R. Franke, *Scattered data interpolation: Tests of some methods*, Math. Comp. **38** (1982), 181–199.

[9] D. Gilberg and N. S. Trudinger, *Elliptic Partial Differental Equations of Second Order*, Springer Verlag, Berlin, 1998, pp. 169–173.

[10] R. L. Hardy, *Multiquadric equations of topography and other irregular surfaces*, J. Geophys. Res. **76** (1971), 1905–1915.

[11] ———, *Theory and applications of the multiquadric-biharmonic method*, Comput. Math. Appl. **19** (1990), 163–208.

[12] Y. Hassan, T. Blanchat, and C. Seeley, Jr., *Simultaneous velocity measurements of both components of a two phase flow using particle image velocimetry*, Int. J. Multiphase Flow **18** (1992), 1–23.

[13] K. Jetter, J. Stöckler, and J. D. Ward, *Error estimates for scattered data interpolation on spheres*, Math. Comp. **68** (1999), 733–747.

[14] ———, *Norming sets and spherical curbature formulas*, Advanced Computational Mathematics: Proceedings of the Guangzhou International Symposium (Z. Chen, Y. Li, Y. Xu, and C. A. Micchelli, eds.), Marcel Dekker, Inc., New York, 1999, pp. 237–244.

[15] E. J. Kansa, *Multiquadrics–a scattered data approximation scheme with applications to computational fluid dynamics–I. Surface approximations and partial derivative estimates*, Computers Math. Applic. **19** (1990), 127–145.

[16] ———, *Multiquadrics–a scattered data approximation scheme with applications to computational fluid dynamics–II. Solutions to parabolic, hyperbolic, and elliptic partial differential equations*, Computers Math. Applic. **19** (1990), 147–161.

[17] W. R. Madych and S. A. Nelson, *Multivariate interpolation and conditionally positive definite functions*, Approx. Theory Appl. **4** (1988), 77–79.

[18] ———, *Multivariate interpolation and conditionally positive definite functions II*, Math. Comp. **54** (1990), 211–230.

[19] C. A. Micchelli, *Interpolation of scattered data: Distance matrices and conditionally positive definite functions*, Constr. Approx. **2** (1986), 11–22.

[20] F. J. Narcowich and J. D. Ward, *Norms of inverses and condition numbers for matrices associated with scattered data*, J. Approx. Theory **64** (1991), 69–94.

[21] ______, *Norm estimates for the inverses of a general class of scattered-data radial-function interpolation matrices*, J. Approx. Theory **69** (1992), 84–109.

[22] ______, *Generalized Hermite interpolation via matrix-valued conditionally positive definite functions*, Math. Comp. **63** (1994), 661–687.

[23] F. J. Narcowich, J. D. Ward, and H. Wendland, *Refined error estimates for radial basis function interpolation*, preprint (2001).

[24] M. J. D. Powell, *The theory of radial basis approximation in 1990*, Advances in Numerical Analysis II: Wavelets, Subdivision and Radial Basis Functions (W. A. Light, ed.), Clarendon Press, Oxford, 1992, pp. 105–210.

[25] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comput. Math. **3** (1995), 251–264.

[26] F. Treves, *Topological Vector Spaces, Distributions, and Kernels*, Academic Press, New York, 1967, p. 287.

[27] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Adv. Comp. Math. **4** (1995), 389–396.

[28] ______, *Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger*, Ph.d. Dissertation, Georg-August University of Göttingen, Germany, 1996.

[29] ———, *Error estimates for interpolation by compactly supported radial basis functions of minimal degree*, J. Approx. Theory **93** (1998), 258–272.

[30] Z. Wu and R. Schaback, *Local error estimates for radial basis function interpolation of scattered data*, IMA J. Numer. Anal. **13** (1993), 13–27.

APPENDIX A

PROOF OF LEMMA 6.4

Proof. We first investigate $|\alpha| = 0$. We have two cases, $j = k$ and $j \neq k$. In order to get upper estimates we will apply Lemma 6.3 repeatedly. If $j = k$, we have

$$\begin{aligned}
|P_{jj}(x)| &\leq c_\gamma \frac{p}{\gamma^2} \sum_{l \neq j} \prod_{t \neq l} \underbrace{|\mathrm{sinc}(x(t)/\gamma)|^p}_{\leq \min\{1, 2\gamma/|x(t)|\}^p} \\
&\quad \cdot \Big\{ (p-1) \underbrace{|\mathrm{sinc}(x(l)/\gamma)|^{p-2} \left|\mathrm{sinc}'(x(l)/\gamma)\right|^2}_{\leq \min\{1, 2\gamma/|x(l)|\}^p} + \underbrace{|\mathrm{sinc}(x(l)/\gamma)|^{p-1} \left|\mathrm{sinc}''(x(l)/\gamma)\right|}_{\leq \min\{1, 2\gamma/|x(l)|\}^p} \Big\} \\
&\leq c_\gamma (s-1) \left(\frac{p}{\gamma}\right)^2 \prod_{t=1}^{s} \min\{1, 2\gamma/|x(t)|\}^p.
\end{aligned}$$

If $j \neq k$, we obtain the following:

$$\begin{aligned}
|P_{jk}(x)| &\leq c_\gamma \left(\frac{p}{\gamma}\right)^2 \prod_{t \neq j,k} \underbrace{|\mathrm{sinc}(x(t)/\gamma)|^p}_{\leq \min\{1, 2\gamma/|x(t)|\}^p} \prod_{r = j,k} \underbrace{|\mathrm{sinc}(x(r)/\gamma)|^{p-1} \left|\mathrm{sinc}'(x(r)/\gamma)\right|}_{\leq \min\{1, 2\gamma/|x(r)|\}^p} \\
&\leq c_\gamma \left(\frac{p}{\gamma}\right)^2 \prod_{t=1}^{s} \min\{1, 2\gamma/|x(t)|\}^p.
\end{aligned}$$

Using the definition $\|A\|_\infty = \max_r \sum_s |a_{rs}|$ for a matrix A, we get

$$\begin{aligned}
&\|\{-\Delta I + \nabla \nabla^T\} \chi_\gamma(x)\|_\infty \\
&\quad \leq c_\gamma (s-1) \left(\frac{p}{\gamma}\right)^2 \prod_{t=1}^{s} \min\{1, 2\gamma/|x(t)|\}^p + (s-1) c_\gamma \left(\frac{p}{\gamma}\right)^2 \prod_{t=1}^{s} \min\{1, 2\gamma/|x(t)|\}^p \\
&\quad = 2 c_\gamma (s-1) \left(\frac{p}{\gamma}\right)^2 \prod_{t=1}^{s} \min\{1, 2\gamma/|x(t)|\}^p.
\end{aligned}$$

This gives the estimate of Lemma 6.4 for $|\alpha| = 0$. Let us now investigate the first order partial derivatives, i.e. $|\alpha| = 1$. We essentially have four different cases. All other cases are obtained by symmetry arguments. The cases are $i = j = k$, $i \neq j = k$, $i = j, j \neq k$, and $i \neq j, k, j \neq k$.

If $i = j = k$, a short calculation gives

$$
\begin{aligned}
\partial_{x(j)} P_{jj}(x) &= -c_\gamma \frac{p^2}{\gamma^3} \sum_{l \neq j} \prod_{t \neq l,j} \mathrm{sinc}^p(x(t)/\gamma) \mathrm{sinc}^{p-1}(x(j)/\gamma) \mathrm{sinc}'(x(j)/\gamma) \\
&\quad \cdot \Big\{ (p-1) \mathrm{sinc}^{p-2}(x(l)/\gamma) \mathrm{sinc}'(x(l)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(l)/\gamma) \mathrm{sinc}''(x(l)/\gamma) \Big\}.
\end{aligned}
$$

This leads to the estimate

$$
\begin{aligned}
|\partial_{x(j)} P_{jj}(x)| &\leq c_\gamma \frac{p^2}{\gamma^3} \sum_{l \neq j} \prod_{t \neq l,j} \underbrace{\mathrm{sinc}^p(x(t)/\gamma)}_{\leq \min\{1, 2\gamma/|x(t)|\}^p} \underbrace{\mathrm{sinc}^{p-1}(x(j)/\gamma) \mathrm{sinc}'(x(j)/\gamma)}_{\leq \min\{1, 2\gamma/|x(j)|\}^p} \\
&\quad \cdot \Big\{ (p-1) \underbrace{\mathrm{sinc}^{p-2}(x(l)/\gamma) \mathrm{sinc}'(x(l)/\gamma)^2}_{\leq \min\{1, 2\gamma/|x(l)|\}^p} + \underbrace{\mathrm{sinc}^{p-1}(x(l)/\gamma) \mathrm{sinc}''(x(l)/\gamma)}_{\leq \min\{1, 2\gamma/|x(l)|\}^p} \Big\} \\
&\leq c_\gamma (s-1) \left(\frac{p}{\gamma} \right)^3 \prod_{t=1}^{s} \min\{1, 2\gamma/|x(t)|\}^p.
\end{aligned}
$$

Some short calculation yields the following remaining first order partial derivatives. If $i \neq j = k$, then

$$
\begin{aligned}
\partial_{x(i)} P_{jj}(x) &= -c_\gamma \frac{p^2}{\gamma^3} \sum_{l \neq j,i} \prod_{t \neq l,i} \mathrm{sinc}^p(x(t)/\gamma) \mathrm{sinc}^{p-1}(x(i)/\gamma) \mathrm{sinc}'(x(i)/\gamma) \\
&\quad \cdot \Big\{ (p-1) \mathrm{sinc}^{p-2}(x(l)/\gamma) \mathrm{sinc}'(x(l)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(l)/\gamma) \mathrm{sinc}''(x(l)/\gamma) \Big\} \\
&\quad - c_\gamma \frac{p}{\gamma^3} \prod_{t \neq i} \mathrm{sinc}^p(x(t)/\gamma) \Big\{ (p-1)(p-2) \mathrm{sinc}^{p-3}(x(i)/\gamma) \mathrm{sinc}'(x(i)/\gamma)^3 \\
&\quad + 3(p-1) \mathrm{sinc}^{p-2}(x(i)/\gamma) \mathrm{sinc}'(x(i)/\gamma) \mathrm{sinc}''(x(i)/\gamma) \\
&\quad + \mathrm{sinc}^{p-1}(x(i)/\gamma) \mathrm{sinc}'''(x(i)/\gamma) \Big\}.
\end{aligned}
$$

If $i = j$ and $j \neq k$, we obtain the following partial derivative:

$$
\begin{aligned}
\partial_{x(j)} P_{jk}(x) &= c_\gamma \frac{p^2}{\gamma^3} \prod_{t \neq j,k} \mathrm{sinc}^p(x(t)/\gamma) \mathrm{sinc}^{p-1}(x(k)/\gamma) \mathrm{sinc}'(x(k)/\gamma) \\
&\quad \cdot \Big\{ (p-1) \mathrm{sinc}^{p-2}(x(j)/\gamma) \mathrm{sinc}'(x(j)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(j)/\gamma) \mathrm{sinc}''(x(j)/\gamma) \Big\}.
\end{aligned}
$$

If $i \neq j, k$, and $j \neq k$, then the partial has the form

$$
\partial_{x(i)} P_{jk}(x) = c_\gamma \left(\frac{p}{\gamma} \right)^3 \prod_{t \neq j,k,i} \mathrm{sinc}^p(x(t)/\gamma) \prod_{r = j,k,i} \mathrm{sinc}^{p-1}(x(r)/\gamma) \mathrm{sinc}'(x(r)/\gamma).
$$

Applying the upper bound for $|\mathrm{sinc}^{(l)}|$ from Lemma 6.3 to these partial derivatives and using the definition of the spectral norm of a matrix give the desired estimates of the form (6.4) for $|\alpha| = 1$.

For $|\alpha| = 2$, we have eight different cases. If $j = k$, then $l = i = j$, $l \neq i \neq j$, $l \neq i = j$, and $l = i \neq j$. If $j \neq k$, then $l \neq i \neq j, k$, $l = i \neq j, k$, $l \neq i = j$, and $l = i = j$. Symmetry arguments cover all remaining cases.

The second order partial derivatives and the upper bounds for the derivatives can be calculated in a similar way as in the case of the partial derivatives of order zero and one. We obtain the following derivatives:

If $l = i = j = k$, then we obtain

$$\begin{aligned}
&\partial^2_{x(j)} P_{jj}(x) \\
&\quad = \quad - c_\gamma \frac{p^2}{\gamma^4} \sum_{s \neq j} \prod_{t \neq s, j} \mathrm{sinc}^p(x(t)/\gamma) \Big\{ (p-1)\mathrm{sinc}^{p-2}(x(j)/\gamma)\mathrm{sinc}'(x(j)/\gamma)^2 \\
&\qquad\qquad + \mathrm{sinc}^{p-1}(x(j)/\gamma)\mathrm{sinc}''(x(j)/\gamma) \Big\} \cdot \Big\{ (p-1)\mathrm{sinc}^{p-2}(x(s)/\gamma)\mathrm{sinc}'(x(s)/\gamma)^2 \\
&\qquad\qquad + \mathrm{sinc}^{p-1}(x(s)/\gamma)\mathrm{sinc}''(x(s)/\gamma) \Big\}.
\end{aligned}$$

If $l \neq i \neq j$, $l \neq j$, and $j = k$, then

$$\begin{aligned}
&\partial_{x(l)} \partial_{x(i)} P_{jj}(x) \\
&\quad = \quad - c_\gamma \frac{p^3}{\gamma^4} \sum_{s \neq j, i} \prod_{t \neq s, i, l} \mathrm{sinc}^p(x(t)/\gamma) \prod_{r = l, i} \mathrm{sinc}^{p-1}(x(r)/\gamma)\mathrm{sinc}'(x(r)/\gamma) \\
&\qquad\qquad \cdot \Big\{ (p-1)\mathrm{sinc}^{p-2}(x(s)/\gamma)\mathrm{sinc}'(x(s)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(s)/\gamma)\mathrm{sinc}''(x(s)/\gamma) \Big\} \\
&\qquad\quad - c_\gamma \frac{p^2}{\gamma^4} \prod_{t \neq l, i} \mathrm{sinc}^p(x(t)/\gamma)\mathrm{sinc}^{p-1}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma) \Big\{ (p-1)(p-2) \\
&\qquad\qquad \cdot \mathrm{sinc}^{p-1}(x(l)/\gamma)\mathrm{sinc}'(x(l)/\gamma)^3 + 3(p-3)\mathrm{sinc}^{p-2}(x(l)/\gamma) \\
&\qquad\qquad \cdot \mathrm{sinc}'(x(l)/\gamma)\mathrm{sinc}''(x(l)/\gamma) + \mathrm{sinc}^{p-1}(x(l)/\gamma)\mathrm{sinc}'''(x(l)/\gamma) \Big\} \\
&\qquad\quad - c_\gamma \frac{p^2}{\gamma^4} \prod_{t \neq l, i} \mathrm{sinc}^p(x(t)/\gamma)\mathrm{sinc}^{p-1}(x(l)/\gamma)\mathrm{sinc}'(x(l)/\gamma) \Big\{ (p-1)(p-2) \\
&\qquad\qquad \cdot \mathrm{sinc}^{p-3}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma)^3 + 3(p-1)\mathrm{sinc}^{p-2}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma) \\
&\qquad\qquad \cdot \mathrm{sinc}''(x(i)/\gamma) + \mathrm{sinc}^{p-1}(x(i)/\gamma)\mathrm{sinc}'''(x(i)/\gamma) \Big\}.
\end{aligned}$$

If $l \neq i = j$, and $j = k$, i.e. $l \neq j$, we get

$$\begin{aligned}
&\partial_{x(l)}\partial_{x(j)}P_{jj}(x) \\
&\quad = \quad - c_\gamma \frac{p^3}{\gamma^4} \sum_{s \neq j,l} \prod_{t \neq s,j,l} \mathrm{sinc}^p(x(t)/\gamma) \prod_{r=l,j} \mathrm{sinc}^{p-1}(x(r)/\gamma)\mathrm{sinc}'(x(r)/\gamma) \\
&\qquad \cdot \Big\{(p-1)\mathrm{sinc}^{p-2}(x(s)/\gamma)\mathrm{sinc}'(x(s)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(s)/\gamma)\mathrm{sinc}''(x(s)/\gamma)\Big\} \\
&\qquad - c_\gamma \frac{p^2}{\gamma^4} \prod_{t \neq l,j} \mathrm{sinc}^p(x(t)/\gamma)\mathrm{sinc}^{p-1}(x(j)/\gamma)\mathrm{sinc}'(x(j)/\gamma) \\
&\qquad \cdot \Big\{(p-1)(p-2)\mathrm{sinc}^{p-3}(x(l)/\gamma)\mathrm{sinc}'(x(l)/\gamma)^3 + 3(p-1)\mathrm{sinc}^{p-2}(x(l)/\gamma) \\
&\qquad \cdot \mathrm{sinc}'(x(l)/\gamma)\mathrm{sinc}''(x(l)/\gamma) + \mathrm{sinc}^{p-1}(x(l)/\gamma)\mathrm{sinc}'''(x(l)/\gamma)\Big\}.
\end{aligned}$$

If $l = i \neq j$, and $j = k$, we obtain

$$\begin{aligned}
&\partial^2_{x(i)}P_{jj}(x) \\
&\quad = \quad - c_\gamma \frac{p^2}{\gamma^4} \sum_{s \neq j,i} \prod_{t \neq s,i} \mathrm{sinc}^p(x(t)/\gamma)\Big\{(p-1)\mathrm{sinc}^{p-2}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma)^2 \\
&\qquad + \mathrm{sinc}^{p-1}(x(i)/\gamma)\mathrm{sinc}''(x(i)/\gamma)\Big\} \cdot \Big\{(p-1)\mathrm{sinc}^{p-2}(x(s)/\gamma)\mathrm{sinc}'(x(s)/\gamma)^2 \\
&\qquad + \mathrm{sinc}^{p-1}(x(s)/\gamma)\mathrm{sinc}''(x(s)/\gamma)\Big\} \\
&\qquad - c_\gamma \frac{p}{\gamma^4} \prod_{t \neq i} \mathrm{sinc}^p(x(t)/\gamma)\Big\{(p-1)(p-2)(p-3)\mathrm{sinc}^{p-4}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma)^4 \\
&\qquad + 6(p-1)(p-2)\mathrm{sinc}^{p-3}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma)^2\mathrm{sinc}''(x(i)/\gamma) \\
&\qquad + 3(p-1)\mathrm{sinc}^{p-2}(x(i)/\gamma)\Big\{\mathrm{sinc}''(x(i)/\gamma)^2 + \mathrm{sinc}'(x(i)/\gamma)\mathrm{sinc}'''(x(i)/\gamma)\Big\} \\
&\qquad + (p-1)\mathrm{sinc}^{p-2}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma)\mathrm{sinc}'''(x(i)/\gamma) \\
&\qquad + \mathrm{sinc}^{p-1}(x(i)/\gamma)\mathrm{sinc}''''(x(i)/\gamma)\Big\}.
\end{aligned}$$

If $l \neq i \neq j, k$ and $j \neq k$, we get

$$\partial_{x(l)}\partial_{x(i)}P_{jk}(x) = c_\gamma \left(\frac{p}{\gamma}\right)^4 \prod_{t \neq j,k,i,l} \mathrm{sinc}^p(x(t)/\gamma) \prod_{r=j,k,i,l} \mathrm{sinc}^{p-1}(x(r)/\gamma)\mathrm{sinc}'(x(r)/\gamma).$$

If $l = i \neq j, k$ and $j \neq k$, we get

$$\begin{aligned}
\partial^2_{x(i)}P_{jk}(x) \quad = \quad & c_\gamma \frac{p^3}{\gamma^4} \prod_{t \neq j,k,i} \mathrm{sinc}^p(x(t)/\gamma) \prod_{r=j,k} \mathrm{sinc}^{p-1}(x(r)/\gamma)\mathrm{sinc}'(x(r)/\gamma)\Big\{(p-1) \\
& \cdot \mathrm{sinc}^{p-2}(x(i)/\gamma)\mathrm{sinc}'(x(i)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(i)/\gamma)\mathrm{sinc}''(x(i)/\gamma)\Big\}.
\end{aligned}$$

If $l \neq i = j$, $j \neq k$, and $l \neq k$ then

$$\begin{aligned}
&\partial_{x(l)}\partial_{x(j)}P_{jk}(x) \\
&\quad = c_\gamma \frac{p^3}{\gamma^4} \prod_{t\neq j,k,l} \mathrm{sinc}^p(x(t)/\gamma) \prod_{r=k,l} \mathrm{sinc}^{p-1}(x(r)/\gamma)\mathrm{sinc}'(x(r)/\gamma) \\
&\qquad \cdot \Big\{(p-1)\mathrm{sinc}^{p-2}(x(j)/\gamma)\mathrm{sinc}'(x(j)/\gamma)^2 + \mathrm{sinc}^{p-1}(x(j)/\gamma)\mathrm{sinc}''(x(j)/\gamma)\Big\}.
\end{aligned}$$

And finally, if $l = i = j$, $j \neq k$, and $j \neq k$, we obtain

$$\begin{aligned}
\partial^2_{x(j)}P_{jk}(x) &= c_\gamma \frac{p^2}{\gamma^4} \prod_{t\neq j,k} \mathrm{sinc}^p(x(t)/\gamma)\mathrm{sinc}^{p-1}(x(k)/\gamma)\mathrm{sinc}'(x(k)/\gamma) \\
&\quad \cdot \Big\{(p-1)(p-2)\mathrm{sinc}^{p-3}(x(j)/\gamma)\mathrm{sinc}'(x(j)/\gamma)^3 \\
&\quad + 3(p-1)\mathrm{sinc}^{p-2}(x(j)/\gamma)\mathrm{sinc}'(x(j)/\gamma)\mathrm{sinc}''(x(j)/\gamma) \\
&\quad + \mathrm{sinc}^{p-1}(x(j)/\gamma)\mathrm{sinc}'''(x(j)/\gamma)\Big\}.
\end{aligned}$$

For all eight cases a calculation in a similar manner as above gives us the bound as stated in Lemma 6.4 for the case $|\alpha| = 2$. This completes the proof. $\square$

APPENDIX B

MATLAB PROGRAM

Main Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                       %
%  Program Name: navier_stokesC8.m                                      %
%                                                                       %
%  This program calculates an interpolation employing RBFs of a        %
%  solution of a Navier Stokes Equation for an incompressible          %
%  fluid flow that is time-independent and has constant pressure.      %
%                                                                       %
%  Needed Files: h4function.m, Lh4function.m, LLh4function.m,          %
%                invers_r_vec.m, Dxh4function.m, Dyh4function.m,       %
%                DxLh4function.m, DyLh4function.m                      %
%                                                                       %
%  Programmer: Svenja Lowitzsch                                         %
%  Contact: lowitzsc@math.tamu.edu                                      %
%                                                                       %
%  Date: May 23, 2001                                                   %
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear all;  close all;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                       %
%                               Input                                   %
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%-----------------------------------------------------------------------%
%nu = 0.002;                                             %critical value%

% viscosity coefficient                                                 %
nu = 1;
```

```
% length and depth, l and d respectively, of cavity                  %
l  = 1;
d  = 1;

% # of steps in x- and y-direction                                    %
hsteps_x = 10;
hsteps_y = hsteps_x;

% # of steps of iteration                                             %
nsteps = 20;
%---------------------------------------------------------------------%

dx = l/hsteps_x;                        % step size for x-direction %
dy = d/hsteps_y;                        % step size for y-direction %


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                     %
%         Determination of grids [X,Y], [XI,YI], [XB,YB]              %
%                                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Determination of grid with edges (0,0), (l,0), (0,-d), (l,-d)       %

%---------------------------------------------------------------------%
[X,Y] = meshgrid(0:dx:l, 0:-dy:-d);
[m,n] = size(X);
%---------------------------------------------------------------------%


% matrices XI, YI of interior points of X, Y (resp.) '(mI)x(mI)'      %

%---------------------------------------------------------------------%
XI = X(2:m-1,2:n-1);                 % subset of INTERIOR points of X   %
YI = Y(2:m-1,2:n-1);                 % subset of INTERIOR points of Y   %
[mI,nI] = size(XI);                  % (mI = (m-1)^2 = m^2-4*m+4)       %
%---------------------------------------------------------------------%


% matrices XB, YB of boundary points of X, Y (resp.) '(mB)x(mB)'      %

%---------------------------------------------------------------------%
```

```
XB_vec = [X(1,1:n-1),X(1:m-1,n)',X(2:m,1)',X(m,2:n)]';
YB_vec = [Y(1,1:n-1),Y(1:m-1,n)',Y(2:m,1)',Y(m,2:n)]';
mBnB   = length(XB_vec);          % (mB = 4(m-1) = 4*m-4)          %
%---------------------------------------------------------------------


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                     %
%  Matrices with all combinations of XI_i-XI_j, XB_i-XB_j,  and       %
%                           XI_i-XB_j                                 %
%                                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Matrix with all combinations of X_i-X_j                             %


X_vec     = reshape(X',1,m*n)';    % changes matrix X to vector X_vec %
Y_vec     = reshape(Y',1,m*n)';    % changes matrix Y to vector Y_vec %


% Matrix with all combinations of XI_i-XI_j and XB_i-XB_j             %


% For [XI,YI], obtain [XXI,YYI] '(mI^2)x(nI^2)' (mI = m-4*m+4):       %

XI_vec    = reshape(XI',1,mI*nI)'; % changes matrix X to vector X_vec %
YI_vec    = reshape(YI',1,mI*nI)'; % changes matrix Y to vector Y_vec %
[XXI,YYI] = meshgrid(XI_vec,YI_vec); % grid: all combis of XI,YI      %
%---------------------------------------------------------------------%
XXI = XXI' - XXI;
YYI = YYI - YYI';
%---------------------------------------------------------------------%
XI = [];  YI = [];


% For [XB,YB], obtain [XXB,YYB] '(mB^2)x(nB^2)':                      %

[XXB,YYB] = meshgrid(XB_vec,YB_vec); % grid: all combis of XB,YB      %
%---------------------------------------------------------------------%
XXB = XXB' - XXB;
YYB = YYB - YYB';
%---------------------------------------------------------------------%
```

```
% Matrix with all combinations of XI_i-XB_j                           %


% For [XB,YB] and [XI,YI], obtain [XXIB,YYIB] '(mI*nI)x(mB*nB)':      %

A    = repmat(XI_vec,1,mBnB);
B    = repmat(XB_vec',mI*nI,1);
%---------------------------------------------------------------------%
XXIB = A - B;
%---------------------------------------------------------------------%
A = [];  B = [];

C    = repmat(YI_vec,1,mBnB);
D    = repmat(YB_vec',mI*nI,1);
%---------------------------------------------------------------------%
YYIB = C - D;
%---------------------------------------------------------------------%
C = [];  D = [];



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                     %
%      Matrices with all combinations of X_i-XI_j and X_i-XB_j        %
%                                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Matrix with all combinations of X_i-XI_j                            %


% For [X,Y] and [XI,YI], obtain [XX2I,YY2I] '(m*n)x(mI*nI)':          %

A    = repmat(X_vec,1,mI*nI);
B    = repmat(XI_vec',m*n,1);
%---------------------------------------------------------------------%
XX2I = A - B;
%---------------------------------------------------------------------%
A = [];  B = [];

C    = repmat(Y_vec,1,mI*nI);
D    = repmat(YI_vec',m*n,1);
%---------------------------------------------------------------------%
```

```
YY2I = C - D;
%-------------------------------------------------------------------%
C = [];  D = [];


% Matrix with all combinations of X_i-XB_j                          %


% For [X,Y] and [XB,YB], obtain [XX2B,YY2B] '(m*n)x(mB*nB)':        %

A    = repmat(X_vec,1,mBnB);
B    = repmat(XB_vec',m*n,1);
%-------------------------------------------------------------------%
XX2B = A - B;
%-------------------------------------------------------------------%
A = [];  B = [];

C    = repmat(Y_vec,1,mBnB);
D    = repmat(YB_vec',m*n,1);
%-------------------------------------------------------------------%
YY2B = C - D;
%-------------------------------------------------------------------%
C = [];  D = [];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                   %
%  Determination of matrices H, LH, LLH for interpolation matrix A  %
%         and determination of matrices DxH and DyH for RHS         %
%                                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Determination of function H, Laplacian LH, and Laplacian squared  %
% LLH for interpolation matrix A in loop '(m^2)x(n^2)'              %

%-------------------------------------------------------------------%
H   = h4function(XXB,YYB);                % '(mB^2)x(nB^2)'        %
LH  = Lh4function(XXIB,YYIB);             % '(mI^2)x(nB^2)'        %
LLH = LLh4function(XXI,YYI);              % '(mI^2)x(nI^2)'        %
%-------------------------------------------------------------------%


% Determination of the partial derivatives DxH, DyH, and DxLH, DyLH  %
% for the right hand side RHSI_r in loop '(1)x(2*m^2)'              %
```

```
%-------------------------------------------------------------------%
DxH  = Dxh4function(XXIB,YYIB);
DyH  = Dyh4function(XXIB,YYIB);
DxLH = DxLh4function(XXI,YYI);
DyLH = DyLh4function(XXI,YYI);
%-------------------------------------------------------------------%
XXI  = [];  YYI  = [];
XXIB = [];  YYIB = [];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                   %
%                    Determination of RHSB                          %
%                                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determination of RHSB = [UB,VB] for initial interpolation           %

%-------------------------------------------------------------------%
UB_vec = zeros(mBnB,1);
UB_vec(1:n,1) = 1;
VB_vec = zeros(mBnB,1);
RHSB_vec =[UB_vec', VB_vec']';
%-------------------------------------------------------------------%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                   %
%   Determination of matrices H2 and LH2 for flow matrices C and D    %
%                                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determination of function H, Laplacian LH, for interpolation        %
% matrices C and D in loop '(m*n)x(m*n)'                              %

%-------------------------------------------------------------------%
H2  = h4function(XX2B,YY2B);                % '(mB*nB)x(m*n)'       %
LH2 = Lh4function(XX2I,YY2I);               % '(mI*nI)x(m*n)'       %
%-------------------------------------------------------------------%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                     %
%    Determination of matrices B, C, and D for plots of velocity      %
%                                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% B*alpha_r = s_r gives interpolated flow after r time steps           %

%---------------------------------------------------------------------%
B = h4function(XXB,YYB);
C = H2;
D = [LH2,H2];
%---------------------------------------------------------------------%
XXB = [];  YYB = [];
H2  = [];  LH2 = [];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                     %
%        Determination of matrices E and F for error analysis  2      %
%                                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% F*alpha_r = grad(s_r) gives gradient of interpolated flow after r   %
% time steps                                                          %

DxH2 = Dxh4function(XX2B,YY2B);
DyH2 = Dyh4function(XX2B,YY2B);

DxLH2 = DxLh4function(XX2I,YY2I);
DyLH2 = DyLh4function(XX2I,YY2I);

%---------------------------------------------------------------------%
E = [DxH2+DyH2];
F = [DxLH2+DyLH2,DxH2+DyH2];
%---------------------------------------------------------------------%
DxLH2 = [];  DyLH2 = [];  DxH2 = [];  DyH2 = [];

XX2B = [];  XX2I = [];
YY2B = [];  YY2I = [];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                     %
%             Initial interpolation to obtain alpha0                  %
```

```
%                                                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Interpolation: Calculation of alpha0                                 %

BS = sparse(B);
%----------------------------------------------------------------------%
alpha0 = BS\RHSB_vec;
%----------------------------------------------------------------------%
BS = [];


% Error analysis: L2-norm of RHS-difference (interpol. - orig. RHS0)  %

%----------------------------------------------------------------------%
norm_0 = norm(B*alpha0 - RHSB_vec)
%----------------------------------------------------------------------%


% Preparation of interpolated RHSB0 and RHS0 for plot                  %
% -> RHSB0=[UB0,VB0] on [XB,YB] and RHS00=[U00,V00] on [X,Y]           %

RHSB0_vec = B*alpha0;                              % on boundary values %
%----------------------------------------------------------------------%
UB0_vec = reshape(RHSB0_vec(1:mBnB),mBnB,1)';
VB0_vec = reshape(RHSB0_vec(mBnB+1:2*mBnB),mBnB,1)';
%----------------------------------------------------------------------%
RHSB0_vec = [];

RHS0_vec = C*alpha0;                                    % on whole grid %
%----------------------------------------------------------------------%
U0 = reshape(RHS0_vec(1:m*n),n,m)';
V0 = reshape(RHS0_vec(m*n+1:2*m*n),n,m)';
%----------------------------------------------------------------------%
RHS0_vec = [];

XI_vec = [];
YI_vec = [];



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                      %
%                   Setup for interpolation loop                       %
```

```
%                                                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%----------------------------------------------------------------------%
%alpha_r = alpha0;
U_r = U0;
V_r = V0;
%----------------------------------------------------------------------%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                      %
%               nsteps interpolations to obtain alpha_k                %
%                                                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


for r = 1:nsteps

   iteration_step = r

   if (r==1)
       alpha_older = alpha0;
       alpha_old   = alpha0;
       %alpha0 = [];

       D_older = C;
       D_old   = C;
       C = [];

       F_older = E;
       F_old   = E;
       E = [];
   else
       alpha_older = alpha_old;
       alpha_old   = alpha_r;
       alpha_r = [];

       D_older = D_old;
       D_old   = D;

       F_older = F_old;
       F_old   = F;
   end
```

```
% Determination of A_r                                              %

%-------------------------------------------------------------------%
UI_r = U_r(2:m-1,2:n-1);        % subset of INTERIOR values of U   %
VI_r = V_r(2:m-1,2:n-1);        % subset of INTERIOR values of V   %
%-------------------------------------------------------------------%

% Interpolation matrix A_r for loop                                 %

UI_r_vec = reshape(UI_r',1,mI*nI)';
VI_r_vec = reshape(VI_r',1,mI*nI)';

UI_r_mat1 = repmat(UI_r_vec,2,2*mI*nI);
VI_r_mat1 = repmat(VI_r_vec,2,2*mI*nI);

UI_r_mat2 = repmat(UI_r_vec,2,2*mBnB);
VI_r_mat2 = repmat(VI_r_vec,2,2*mBnB);

A11_r = nu*LLH-(UI_r_mat1.*DxLH+VI_r_mat1.*DyLH);
A12_r = nu*LH -(UI_r_mat2.*DxH+VI_r_mat2.*DyH);

%-------------------------------------------------------------------%
A_r = [A11_r,A12_r; LH',H];
%-------------------------------------------------------------------%

UI_r_vec  = [];  VI_r_vec  = [];
UI_r_mat1 = [];  VI_r_mat1 = [];
UI_r_mat2 = [];  VI_r_mat2 = [];
A11_r = [];  A12_r = [];


% Calculation of RHSI_r                                             %

%-------------------------------------------------------------------%
RHSUI_r = zeros(mI*nI,1);
RHSVI_r = zeros(mI*nI,1);
RHSI_r  = [RHSUI_r,RHSVI_r];
%-------------------------------------------------------------------%
RHSUI_r = [];  RHSVI_r = [];


% Interpolation: Calculation of alpha_r                             %
```

```
%----------------------------------------------------------------------%
[alpha_r,RHS_r_vec] = invers_r_vec(A_r,RHSI_r,RHSB_vec);
%----------------------------------------------------------------------%
A_r = [];


% Preparation of interpolated RHS_r for plot                           %
% and obtain RHS_r = [U_r,V_r] for next step in loop                   %

RHSr_vec = D*alpha_r;                                 % on whole grid %
%----------------------------------------------------------------------%
U_r = reshape(RHSr_vec(1:m*n),n,m)';
V_r = reshape(RHSr_vec(m*n+1:2*m*n),n,m)';
%----------------------------------------------------------------------%
RHSr_vec = [];



% Error analysis I:                                                    %
% L2-norm of {interpol_r - interpol_(r-1)}                             %

%----------------------------------------------------------------------%
norm1_r = norm(D*alpha_r - D_old*alpha_old)
norm1_I(r) = norm1_r;
%----------------------------------------------------------------------%


% Error analysis II:                                                   %
% L2-norm of {interpol_r - interpol_(r-1)}                             %
% + L2-norm of {grad(interpol_r - interpol_(r-1))}                     %

%----------------------------------------------------------------------%
norm2_r = norm(D*alpha_r - D_old*alpha_old) ...
          + norm(F*alpha_r - F_old*alpha_old)
norm2_I(r) = norm2_r;
%----------------------------------------------------------------------%


% Error analysis III:                                                  %
% L2-norm of {interpol_(r-1) - interpol_(r-2)}                         %
% + L2-norm of {grad(interpol_(r-1) - interpol_(r-2))}/                %
% L2-norm of {interpol_r - interpol_(r-1)}                             %
% + L2-norm of {grad(interpol_r - interpol_(r-1))}                     %

a = norm(D_old*alpha_old - D_older*alpha_older);
```

```
b = norm(F_old*alpha_old - F_older*alpha_older);
c = norm(D*alpha_r - D_old*alpha_old);
d = norm(F*alpha_r - F_old*alpha_old);

%-------------------------------------------------------------------%
norm3_r = (a+b)/(c+d)
norm3_I(r) = norm3_r;
%-------------------------------------------------------------------%


% Error analysis IV:                                                %
% L2-norm of {LHS_r - RHS_r}                                        %

% Calculation of LHS_r (interior and exterior data)                 %

% Calculation of LHS_r:                                             %
% Matrices UI_r, VI_r of interior points of U_r, V_r (resp.)        %
% '(mI)x(nI)'                                                       %

%-------------------------------------------------------------------%
UI_r = U_r(2:m-1,2:n-1);        % subset of INTERIOR values of U   %
VI_r = V_r(2:m-1,2:n-1);        % subset of INTERIOR values of V   %
%-------------------------------------------------------------------%

% Calculation of A_r                                                %

UI_r_vec = reshape(UI_r',1,mI*nI)';
VI_r_vec = reshape(VI_r',1,mI*nI)';
UI_r = [];  VI_r = [];

UI_r_mat1 = repmat(UI_r_vec,2,2*mI*nI);
VI_r_mat1 = repmat(VI_r_vec,2,2*mI*nI);

UI_r_mat2 = repmat(UI_r_vec,2,2*mBnB);
VI_r_mat2 = repmat(VI_r_vec,2,2*mBnB);

A11_r = nu*LLH-(UI_r_mat1.*DxLH+VI_r_mat1.*DyLH);
A12_r = nu*LH -(UI_r_mat2.*DxH+VI_r_mat2.*DyH);

%-------------------------------------------------------------------%
A_r = [A11_r,A12_r; LH',H];
%-------------------------------------------------------------------%

UI_r_vec  = [];  VI_r_vec  = [];
UI_r_mat1 = [];  VI_r_mat1 = [];
```

```
   UI_r_mat2 = [];  VI_r_mat2 = [];
   A11_r = [];  A12_r = [];

   LHS_r_vec = A_r*alpha_r;
   A_r = [];

   norm4_r = norm(LHS_r_vec - RHS_r_vec)
   %-------------------------------------------------------------------%
   norm4_I(r) = norm4_r;
   %-------------------------------------------------------------------%
   LHS_r_vec = [];


end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                     %
%                             Output                                  %
%                                                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

norm0 = norm_0
norm1 = norm1_I
norm2 = norm2_I
norm3 = norm3_I
norm4 = norm4_I

% Original initial data on boundary                                   %

quiver(XB_vec,YB_vec,UB_vec,VB_vec);


% Interpolated initial data on boundary                               %

figure;
quiver(XB_vec,YB_vec,UB0_vec,VB0_vec);


% Interpolated initial data                                           %

figure;
quiver(X,Y,U0,V0);
```

```
% Interpolated steady state data after nsteps steps                    %

figure;
quiver(X,Y,U_r,V_r);


H = [];  LH = [];  LLH = [];
DxH = [];  DyH = [];  DxLH = [];  DyLH = [];
B = [];  D = [];  F = [];
D_old = [];  D_older = [];
F_old = [];  F_older = [];
alpha_old = [];  alpha_older = [];
RHSB = [];  UB = [];  VB = [];
UB0_vec = [];  VB0_vec = [];
RHSB_vec = [];  UB_vec = [];  VB_vec = [];
RHS_r_vec = [];
alpha_r = [];
U_r = [];  V_r = [];
U0  = [];  V0  = [];
X = [];  Y = [];  XB_vec = [];  YB_vec = [];
```

Subroutine for Φ

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                       %
% Program Name: h4function.m                                            %
%                                                                       %
%                                                                       %
% Evaluates 2D matrix-valued divergence-free Wendland function values   %
% H.                                                                    %
%                                                                       %
% Input:  gridded matrix data (X2, Y2) (where X, Y are matrices)        %
%         (all combis)                                                  %
%                                                                       %
% Output: function value matrix H (compare H(i,j) with X(i,j)           %
%         and Y(i,j)from X,Y below!                                     %
%                                                                       %
%                                                                       %
% Programmer: Svenja Lowitzsch                                          %
% Contact: lowitzsc@math.tamu.edu                                       %
%                                                                       %
% Date: May 17, 2001                                                    %
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function H = h4function(X2,Y2)

[m,n] = size(X2);

H11 = ones(m,n);
H22 = ones(m,n);
H12 = ones(m,n);

X = X2;
Y = Y2;

norm2_mat=X.^2+Y.^2;                % squared L2 norm for Xi-Xj, Yi-Yj %
R = sqrt(norm2_mat);                % L2 norm for all Xi-Xj, Yi-Yj     %

T = (R-ones(m,n));

% Note: H11(0) = 26;  H22(0) = 26;  H12(0) = H21(0) = 0;               %
for i = 1:m
    for j = 1:n
       if (norm2_mat(i,j) <= 1)
          H11(i,j) = (-5.2)*(231*(X(i,j)^2+13*Y(i,j)^2)*R(i,j)^2 ...
                            -24*(-41*Y(i,j)^2+3*X(i,j)^2)*R(i,j) ...
                            -114*X(i,j)^2+18*Y(i,j)^2-40*R(i,j) ...
                            -5)*T(i,j)^8;
          H22(i,j) = (-5.2)*(231*(13*X(i,j)^2+Y(i,j)^2)*R(i,j)^2 ...
                            -24*(3*Y(i,j)^2-41*X(i,j)^2)*R(i,j) ...
                            -114*Y(i,j)^2+18*X(i,j)^2-40*R(i,j) ...
                            -5)*T(i,j)^8;
          H12(i,j) = 686.4*(8*R(i,j)+21*R(i,j)^2+1)*X(i,j)*Y(i,j) ...
                            *T(i,j)^8;
       else
          H11(i,j) = 0;
          H22(i,j) = 0;
          H12(i,j) = 0;
        end
    end
end

H = [H11,H12; H12,H22];
```

Subroutine for $\Delta\Phi$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                      %
```

```
% Program Name: Lh4function.m                                          %
%                                                                      %
%                                                                      %
% Evaluates Laplacian of 2D matrix-valued divergence-free Wendland     %
% function values LH  (C_6)                                            %
%                                                                      %
% Input:  gridded matrix data (X2, Y2) (where X2, Y2 are matrices)     %
%         (all combis)                                                 %
%                                                                      %
% Output: Laplacian of function values, matrix LH                      %
%                                                                      %
%                                                                      %
% Programmer: Svenja Lowitzsch                                         %
% Contact: lowitzsc@math.tamu.edu                                      %
%                                                                      %
% Date: May 17, 2001                                                   %
%                                                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function LH = Lh4function(X2,Y2)

[m,n] = size(X2);

LH11 = ones(m,n);
LH22 = ones(m,n);
LH12 = ones(m,n);

X = X2;
Y = Y2;

norm2_mat = X.^2+Y.^2;             % squared L2 norm for Xi-Xj, Yi-Yj %
R = sqrt(norm2_mat);               % L2 norm for all Xi-Xj, Yi-Yj     %

T = (R-ones(m,n));

% Note: LH11(0) = -2745.6, LH22(0) = -2745.6, LH12(0) = 0 %
for i = 1:m
    for j = 1:n
       if (norm2_mat(i,j) <= 1)
          LH11(i,j) = (-1372.8)*(147*(X(i,j)^2+11*Y(i,j)^2)*R(i,j)^2 ...
                      -2*(349*Y(i,j)^2+79*X(i,j)^2)*R(i,j) ...
                      +12*R(i,j)-3*X(i,j)^2-93*Y(i,j)^2+2) ...
                      *T(i,j)^6;
          LH22(i,j) = (-1372.8)*(147*(11*X(i,j)^2+Y(i,j)^2)*R(i,j)^2 ...
                      -2*(349*X(i,j)^2+79*Y(i,j)^2)*R(i,j) ...
                      +12*R(i,j)-3*Y(i,j)^2-93*X(i,j)^2+2) ...
```

```
                                *T(i,j)^6;
          LH12(i,j) = 41184*(49*R(i,j)^2-18*R(i,j)-3)*X(i,j)*Y(i,j) ...
                                *T(i,j)^6;
        else
          LH11(i,j) = 0;
          LH22(i,j) = 0;
          LH12(i,j) = 0;
        end
    end
end

LH = [LH11,LH12; LH12,LH22];
```

Subroutine for $\Delta^2\Phi$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                         %
% Program Name: LLh4function.m                                            %
%                                                                         %
%                                                                         %
% Evaluates 2D matrix-valued divergence-free Wendland function values     %
% LLH.                                                                    %
%                                                                         %
% Input:  gridded matrix data (X2, Y2) (where X, Y are matrices)          %
%         (all combis)                                                    %
%                                                                         %
% Output: function value matrix LLH (compare H(i,j) with X(i,j)           %
%         and Y(i,j)from X,Y below!                                       %
%                                                                         %
%                                                                         %
% Programmer: Svenja Lowitzsch                                            %
% Contact: lowitzsc@math.tamu.edu                                         %
%                                                                         %
% Date: May 17, 2001                                                      %
%                                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function LLH = LLh4function(X2,Y2)

[m,n] = size(X2);

LLH11 = ones(m,n);
LLH22 = ones(m,n);
LLH12 = ones(m,n);
```

```
X = X2;
Y = Y2;

norm2_mat = X.^2+Y.^2;              % squared L2 norm for Xi-Xj, Yi-Yj %
R = sqrt(norm2_mat);                % L2 norm for all Xi-Xj, Yi-Yj     %

T = (R-ones(m,n));

% Note: LLH11(0) = 494208, LLH22(0) = 494208, LLH12(0) = 0             %
for i = 1:m
    for j = 1:n
       if (norm2_mat(i,j) <= 1) & (norm2_mat(i,j) > 0)
          LLH11(i,j) = (-988416*(23*X(i,j)^2+79*Y(i,j)^2)-24216192 ...
                  *(X(i,j)^2+9*Y(i,j)^2)*R(i,j)^2+494208 ...
                  +1976832*R(i,j)+(1731704833*Y(i,j)^6*X(i,j)^4 ...
                  +1080338688*Y(i,j)^8*X(i,j)^2+436879872*X(i,j)^8 ...
                  *Y(i,j)^2+1302732288*X(i,j)^6*Y(i,j)^4+44478720 ...
                  *X(i,j)^10+258964992*Y(i,j)^10)/(R(i,j)^7))*T(i,j)^4;
          LLH22(i,j) = (-988416*(79*X(i,j)^2+23*Y(i,j)^2) ...
                  -24216192*(Y(i,j)^2+9*X(i,j)^2)*R(i,j)^2+494208 ...
                  +1976832*R(i,j)+(1302732289*Y(i,j)^6*X(i,j)^4 ...
                  +436879872*Y(i,j)^8*X(i,j)^2+1080338688*X(i,j)^8 ...
                  *Y(i,j)^2+1731704832*X(i,j)^6*Y(i,j)^4+258964992 ...
                  *X(i,j)^10+44478720*Y(i,j)^10)/(R(i,j)^7))*T(i,j)^4;
          LLH12(i,j) = 6918912*(28*R(i,j)^2-31*R(i,j)+8)*X(i,j) ...
                       *Y(i,j)*T(i,j)^4;
        elseif  (norm2_mat(i,j) == 0)
           LLH11(i,j) = 494208;
           LLH22(i,j) = 494208;
           LLH12(i,j) = 0;
        else
           LLH11(i,j) = 0;
           LLH22(i,j) = 0;
           LLH12(i,j) = 0;
        end
    end
 end

 LLH = [LLH11,LLH12; LLH12,LLH22];
```

Subroutine for A^{-1} in Iteration

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                       %
```

```
% Program Name: invers_r_vec.m                                      %
%                                                                   %
% Evaluates 2D matrix-valued divergence-free Wendland interpolation %
% coefficients alpha.                                               %
%                                                                   %
% It first evaluates the interpolation matrix A, then turns it into %
% the sparse matrix B, and then solves the system B*alpha=vec (RHS) %
% for alpha.                                                        %
%                                                                   %
% Input:  RBF interpolation matrix H                                %
%         function values RHSI_r for interior                       %
%         function values RHSB_vec for boundary                     %
%                                                                   %
% Output: solution coefficient vector alpha                         %
%                                                                   %
%                                                                   %
% Programmer: Svenja Lowitzsch                                      %
% Contact: lowitzsc@math.tamu.edu                                   %
%                                                                   %
% Date: March 07, 2001                                              %
%                                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [alpha,RHS_r_vec] = invers_r_vec(A,RHSI_r,RHSB_vec)

[MI,NI] = size(RHSI_r);

% changes matrix RHSI_r to vector RHSI_r_vec %
UI = reshape(RHSI_r(:,1:NI/2)',MI*NI/2,1);
VI = reshape(RHSI_r(:,NI/2+1:NI)',MI*NI/2,1);
RHSI_r_vec=[UI',VI']';

RHS_r_vec = [RHSI_r_vec', RHSB_vec']';

B = sparse(A);
alpha=B\RHS_r_vec;                          % solves alpha = inv(H)*RHS %
```

Subroutine for $\partial_1\Phi$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                   %
% Program Name: Dxh4function.m                                      %
%                                                                   %
%                                                                   %
% Evaluates 2D matrix-valued divergence-free Wendland function values %
```

```
% DxH.                                                                 %
%                                                                      %
% Input:  gridded matrix data (X2, Y2) (where X, Y are matrices)       %
%         (all combis)                                                 %
%                                                                      %
% Output: function value matrix DxH (compare H(i,j) with X(i,j)        %
%         and Y(i,j)from X,Y below!                                    %
%                                                                      %
%                                                                      %
% Programmer: Svenja Lowitzsch                                         %
% Contact: lowitzsc@math.tamu.edu                                      %
%                                                                      %
% Date: May 17, 2001                                                   %
%                                                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DxH = Dxh4function(X2,Y2)

[m,n] = size(X2);

DxH11 = ones(m,n);
DxH22 = ones(m,n);
DxH12 = ones(m,n);

X = X2;
Y = Y2;

norm2_mat=X.^2+Y.^2;                % squared L2 norm for Xi-Xj, Yi-Yj %
R = sqrt(norm2_mat);                % L2 norm for all Xi-Xj, Yi-Yj     %

T = (R-ones(m,n));

% Note: DxH11(0) = 0;  DxH22(0) = 0;  DxH12(0) = 0;                    %
for i = 1:m
    for j = 1:n
       if (norm2_mat(i,j) <= 1)
          DxH11(i,j) = (-686.4)*(21*(X(i,j)^2+11*Y(i,j)^2)*R(i,j) ...
                                -13*X(i,j)^2+17*Y(i,j)^2-7*R(i,j) ...
                                -1)*X(i,j)*T(i,j)^7;
          DxH22(i,j) = (-2059.2)*(7*(3*Y(i,j)^2+13*X(i,j)^2)*R(i,j) ...
                                -3*X(i,j)^2-13*Y(i,j)^2 ...
                                -7*R(i,j)-1)*X(i,j)*T(i,j)^7;
          DxH12(i,j) = 686.4*(21*(Y(i,j)^2+11*X(i,j)^2)*R(i,j) ...
                                +17*X(i,j)^2-13*Y(i,j)^2 ...
                                -7*R(i,j)-1)*Y(i,j)*T(i,j)^7;
       else
```

```
          DxH11(i,j) = 0;
          DxH22(i,j) = 0;
          DxH12(i,j) = 0;
        end
    end
 end

 DxH = [DxH11,DxH12; DxH12,DxH22];
```

Subroutine for $\partial_2\Phi$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                       %
% Program Name: Dyh4function.m                                          %
%                                                                       %
%                                                                       %
% Evaluates 2D matrix-valued divergence-free Wendland function values  %
% DyH.                                                                  %
%                                                                       %
% Input:  gridded matrix data (X2, Y2) (where X, Y are matrices)       %
%         (all combis)                                                  %
%                                                                       %
% Output: function value matrix DyH (compare H(i,j) with X(i,j)        %
%         and Y(i,j)from X,Y below!                                     %
%                                                                       %
%                                                                       %
% Programmer: Svenja Lowitzsch                                          %
% Contact: lowitzsc@math.tamu.edu                                       %
%                                                                       %
% Date: May 17, 2001                                                    %
%                                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DyH = Dyh4function(X2,Y2)

[m,n] = size(X2);

DyH11 = ones(m,n);
DyH22 = ones(m,n);
DyH12 = ones(m,n);

X = X2;
Y = Y2;

norm2_mat=X.^2+Y.^2;                % squared L2 norm for Xi-Xj, Yi-Yj %
```

```
R = sqrt(norm2_mat);                % L2 norm for all Xi-Xj, Yi-Yj     %

T = (R-ones(m,n));

% Note:  DyH11(0) = 0; DyH22(0) = 0;  DyH12(0) = 0;                    %
for i = 1:m
    for j = 1:n
       if (norm2_mat(i,j) <= 1) & (norm2_mat(i,j) > 0)
          DyH11(i,j) = (-2059.2)*(7*(13*Y(i,j)^2+3*X(i,j)^2)*R(i,j) ...
                                 -13*X(i,j)^2-3*Y(i,j)^2 ...
                                 -7*R(i,j)-1)*Y(i,j)*T(i,j)^7;
          DyH22(i,j) = (-686.4)*(21*(11*X(i,j)^2+Y(i,j)^2)*R(i,j) ...
                                 +17*X(i,j)^2-13*Y(i,j)^2 ...
                                 -7*R(i,j)-1)*Y(i,j)*T(i,j)^7;
          DyH12(i,j) = 686.4*(21*(X(i,j)^2+11*Y(i,j)^2)*R(i,j) ...
                                 -13*X(i,j)^2+17*Y(i,j)^2 ...
                                 -7*R(i,j)-1)*X(i,j)*T(i,j)^7;
       else
          DyH11(i,j) = 0;
          DyH22(i,j) = 0;
          DyH12(i,j) = 0;
       end
    end
 end

 DyH = [DyH11,DyH12; DyH12,DyH22];
```

Subroutine for $\partial_1 \Delta \Phi$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                      %
% Program Name: DxLh4function.m                                        %
%                                                                      %
%                                                                      %
% Evaluates 2D matrix-valued divergence-free Wendland function values  %
% DxLH.                                                                %
%                                                                      %
% Input:  gridded matrix data (X2, Y2) (where X, Y are matrices)       %
%         (all combis)                                                 %
%                                                                      %
% Output: function value matrix DxLH (compare H(i,j) with X(i,j)       %
%         and Y(i,j)from X,Y below!                                    %
%                                                                      %
%                                                                      %
```

```
% Programmer: Svenja Lowitzsch                                      %
% Contact: lowitzsc@math.tamu.edu                                   %
%                                                                   %
% Date: May 17, 2001                                                %
%                                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DxLH = DxLh4function(X2,Y2)

[m,n] = size(X2);

DxLH11 = ones(m,n);
DxLH22 = ones(m,n);
DxLH12 = ones(m,n);

X = X2;
Y = Y2;

norm2_mat=X.^2+Y.^2;                % squared L2 norm for Xi-Xj, Yi-Yj %
R = sqrt(norm2_mat);                % L2 norm for all Xi-Xj, Yi-Yj     %

T = (R-ones(m,n));

% Note: DxLH11(0) = 0;  DxLH22(0) = 0;  DxLH12(0) = 0;               %
for i = 1:m
    for j = 1:n
        if (norm2_mat(i,j) <= 1) & (norm2_mat(i,j) > 0)
            DxLH11(i,j) = (-41184)*(49*(X(i,j)^2+9*Y(i,j)^2)*R(i,j) ...
                                   -291*Y(i,j)^2-67*X(i,j)^2 ...
                                   +15*R(i,j)+3)*X(i,j)*T(i,j)^5;
            DxLH22(i,j) = (-41184)*(49*(3*Y(i,j)^2+11*X(i,j)^2) ...
                                   *R(i,j)-425*X(i,j)^2-201 ...
                                   *Y(i,j)^2+45*R(i,j)+9)*X(i,j) ...
                                   *T(i,j)^5;
            DxLH12(i,j) = 41184*(49*(Y(i,j)^2+9*X(i,j)^2)*R(i,j) ...
                                   -291*X(i,j)^2-67*Y(i,j)^2 ...
                                   +15*R(i,j)+3)*Y(i,j)*T(i,j)^5;
         else
            DxLH11(i,j) = 0;
            DxLH22(i,j) = 0;
            DxLH12(i,j) = 0;
        end
    end
 end

 DxLH = [DxLH11,DxLH12; DxLH12,DxLH22];
```

Subroutine for $\partial_2\Delta\Phi$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                          %
% Program Name: DyLh4function.m                                            %
%                                                                          %
%                                                                          %
% Evaluates 2D matrix-valued divergence-free Wendland function values      %
% DyLH.                                                                    %
%                                                                          %
% Input:  gridded matrix data (X2, Y2) (where X, Y are matrices)           %
%         (all combis)                                                     %
%                                                                          %
% Output: function value matrix DyLH (compare H(i,j) with X(i,j)           %
%         and Y(i,j)from X,Y below!                                        %
%                                                                          %
%                                                                          %
% Programmer: Svenja Lowitzsch                                             %
% Contact: lowitzsc@math.tamu.edu                                          %
%                                                                          %
% Date: May 17, 2001                                                       %
%                                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DyLH = DyLh4function(X2,Y2)

[m,n] = size(X2);

DyLH11 = ones(m,n);
DyLH22 = ones(m,n);
DyLH12 = ones(m,n);

X = X2;
Y = Y2;

norm2_mat=X.^2+Y.^2;                 % squared L2 norm for Xi-Xj, Yi-Yj %
R = sqrt(norm2_mat);                 % L2 norm for all Xi-Xj, Yi-Yj     %

T = (R-ones(m,n));

% Note: DyLH11(0) = 0;  DyLH22(0) = 0;  DyLH12(0) = 0;                     %
for i = 1:m
    for j = 1:n
        if (norm2_mat(i,j) <= 1)
            DyLH11(i,j) = (-41184)*(49*(3*X(i,j)^2+11*Y(i,j)^2) ...
                                  *R(i,j)-425*Y(i,j)^2-201*X(i,j)^2 ...
```

```
                                +45*R(i,j)+9)*Y(i,j)*T(i,j)^5;
           DyLH22(i,j) = (-41184)*(49*(9*X(i,j)^2+Y(i,j)^2)*R(i,j) ...
                                -291*X(i,j)^2-67*Y(i,j)^2 ...
                                +15*R(i,j)+3)*Y(i,j)*T(i,j)^5;
           DyLH12(i,j) = 41184*(49*(X(i,j)^2+9*Y(i,j)^2)*R(i,j) ...
                                -291*Y(i,j)^2-67*X(i,j)^2 ...
                                +15*R(i,j)+3)*X(i,j)*T(i,j)^5;
        else
           DyLH11(i,j) = 0;
           DyLH22(i,j) = 0;
           DyLH12(i,j) = 0;
       end
   end
end

DyLH = [DyLH11,DyLH12; DyLH12,DyLH22];
```

VITA

Svenja Lowitzsch was born in Kiel, Germany on October 22, 1969. She received her Diploma in Mathematics from Georg-August University (Germany) in October 1996. She was a research assistant at the Max-Planck Institute for Experimental Medicine (Germany) from December 1996 to December 1997. She has been a graduate assistant in the Department of Mathematics at Texas A&M University from September 1997 to the present. From June to August 2001 she was an intern at The Boeing Company, Washington. Her permanent address is An den Talwiesen 20, 34225 Baunatal, Germany.

This dissertation was typeset using LaTeX.